Link the head to the "beak": Zero Shot Learning from Noisy Text Description at Part Precision

Mohamed Elhoseiny[1,2*], Yizhe Zhu[1*], Han Zhang[1], and Ahmed Elgammal[1]
elhoseiny@fb.com, yizhe.zhu@rutgers.edu, {han.zhang, elgammal}@cs.rutgers.edu
[1]Rutgers University, Department of Computer Science, [2] Facebook AI Research

Abstract

In this paper, we study learning visual classifiers from unstructured text descriptions at part precision with no training images. We propose a learning framework that is able to connect text terms to its relevant parts and suppress connections to non-visual text terms without any part-text annotations. For instance, this learning process enables terms like "beak" to be sparsely linked to the visual representation of parts like head, while reduces the effect of non-visual terms like "migrate" on classifier prediction. Images are encoded by a part-based CNN that detect bird parts and learn part-specific representation. Part-based visual classifiers are predicted from text descriptions of unseen visual classifiers to facilitate classification without training images (also known as zero-shot recognition). We performed our experiments on CUBirds 2011 dataset and improves the state-of-the-art text-based zero-shot recognition results from 34.7% to 43.6%. We also created large scale benchmarks on North American Bird Images augmented with text descriptions, where we also show that our approach outperforms existing methods. Our code, data, and models are publically available link [1].

1. Introduction

Recognizing visual categories only from the class description is an appealing characteristic of human learning and generalization, which is desirable to be modeled for better machine intelligence. This problem is known as "zero-shot" learning/classification. In practice, this is motivated by the lack of annotated training data for most object categories and especially at the fine-grained level, which has been observed by several researches (e.g., [40, 52]). For instance, there exist tens of thousands of bird categories among which images are available for only few-hundred-categories in existing datasets ($< 5\%$) [48]. Some bird categories are scarce in the real-world– it is very hard to find the "Crested ibis" around us and even in a zoo.

Earlier zero-shot recognition methods rely on describing visual classes by a set of semantically meaningful properties known as attributes [16, 24]. The underlying principle behind the success of attributes on zero-shot learning is that they are modeled as an intermediate layer between class labels and images, which enable transfer of shared concepts/attributes from seen classes to unseen classes. More recent attribute methods improve the information transfer across classes by joint embedding of images and attributes [4, 49, 10, 5]. While attributes can semantically describe classes with human interpretability without any images, they typically require domain experts to be defined. It is also necessary to collect hundreds of these attribute annotations for each of the seen and unseen classes which is discouraging.

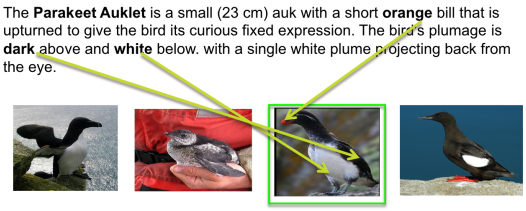

Figure 1. People can learn from text descriptions at part-level

Towards reducing the gap between machine and human intelligence on this task, recent methods [14, 26, 6, 36] explored zero-shot learning from online text descriptions, which in turn avoids the burden of heavy attributes annotations for each class. What makes this setting very challenging is that these descriptions comes in the form of noisy encyclopedia articles that include not only visual descriptions about the visual appearance but also discussion about the category's behavior, breading, immigration, etc. Our work aims at designing an interpretable model in this direction. Prior works [36, 39, 26, 6] use a wholistic feature representation for both the object and the text description (e.g., term frequency vector is common for the bird text description and a visual feature vector for the whole object).

Contributions In our work, we propose an effective model that can relate text information of visual categories to images with part-based regularization. Fig. 1 illustrates the text-part connectivity capability that we aim to model in our work, where birds are recognized from text description by relating text terms to parts in the image (e.g, relating the

* Both authors contributed equally to this work

bill to the head of the bird). Note that this task is unlike existing visual grounding tasks (e.g., [35, 18]), which requires object-(text phrase) annotations during training and has been mainly studied at the object level/not at part-level. Our method is able to quell the noise in the text descriptions by eliminating irrelevant text information without requiring part-text correspondence annotation or part annotations at test time. Our model is composed of two networks, *"Visual Part Detector / Encoder network"* (VPDE-net) and *"Part Zero-Shot Classifier prediction network"* (PZSC-net). The VPDE-net is fed with bird images, detects the bird parts, and learns CNN feature representation for every part. The PZSC-net predicts part-based zero-shot classifier from the noisy text description of bird classes, which is executed on the part-CNN representation produced by the VPDE-net.

Besides evaluating on the CUB dataset [48], we also set up new zero-shot benchmarks by extending the NABirds dataset [43] with a corresponding unstructured text article extracted from Wikipedia and AllaboutBirds website [2]. This is five times bigger than the largest existing benchmark for text-based zero shot learning.

2. Related Work

Attribute-based methods: Besides manually specified attributes (e.g., [25, 16, 24, 33]), several researchers have explored various attribute applications and attempted to automatically discover these attributes [9, 37, 29, 38]. Recent approaches model attributes in a continuous space (e.g., [4, 21]). The main idea of these approaches is to learn a transformation matrix $\mathbf{W}$ that correlates attributes to images –we name these methods *transformation-based approaches*. Other zero-shot approaches used *graph/hyper-graphs* built on attributes and class labels (e.g., [17, 20]). In contrast to graph/hyper-graph based approaches, *transformation-based approaches* have recently shown better performance and are meanwhile simpler and more efficient on fine-grained recognition (e.g., [39, 6, 5]).

Text-based methods: More relevant to this paper is the research direction exploring using text articles from the web to predict zero-shot visual classifiers. Elhoseiny *et al*. [14] proposed an approach to that combines domain transfer and regression to predict visual classifiers from a TF-IDF textual representation. Bo *et al*. [26] adopted deep neural networks to predict convolutional classifiers, leading to a noticeable improvement on zero-shot classification. Very recently, Qiao *et al*. [36] revisited the importance of regularization on zero-shot learning. They show that attribute-based formulation like [39] achieves competitive zero-shot performance when applied to text by just replacing the attribute representation with textual feature vectors. They further demonstrated that the noise in the text descriptions could be suppressed by encouraging group sparsity on the connections to the textual terms. Similar to *transformation-based approaches*, most of these text-based methods (e.g., [14, 36, 39]) are based also on learning transformations that relates images to text in a common space. In our view, most of the recent progress has been achieved by better visual representations using deep neural networks (e.g., [26]) and/or better regularization to suppress noise in texts (e.g. [36, 39]). In our work, we build on top the existing methods and demonstrate that zero-shot recognition could be significantly improved by part-based regularization in contrast to the whole image in the aforementioned approaches. It is important to mention that in [3], Akata *et al*. studied zero-shot learning with multiple cues and they used bird parts. There are two key differences to our work. (1) In [3], multiple sources from WordNet [31] and word embeddings [30, 34] are used in addition to text terms, while we only uses text terms. (2) They used annotations of 19 bird parts for training, however, at test time the method is not able to locate these parts and hence require the part test annotations to relate to their multiple cues. In our work, we demonstrated significantly better performance using only text terms and with no part annotation needed at test time. Moreover, at training time, only annotations of 7 parts are needed instead of 19 that are easier to collect.

Other language& vision methods: In other tasks like image-captioning (e.g., [22, 47, 15]),VQA (e.g., [7]), and image-sentence similarity (e.g., [23, 45]), better performance has been demonstrated with better image and language representations. The text annotations in the typical datasets for these methods are carefully collected at the image-level by crowdsourcing services (e.g., 5 captions per sentences in MS-COCO [28] or Flick30K datasets [50]). In contrast to these settings, the text descriptions in our work come at the category level (e.g., one text description for "Cardinal" class). Hence, there is much less text in our setting and meanwhile the text is much noisier as we described earlier. In our experiments, we set up an image-sentence similarity baseline to study the performance of the representations in methods when applied to very noisy text as in our setting with only the small portion of the text is related visually.

3. Proposed Approach

Connecting unstructured text into bird parts requires language and a visual representations that facilitates mutual transfer at the part level from text to images and vice versa. We also aim at a formulation that does not require text-to-part labeling at training time nor it does require oracle part annotations at test time (e.g., [3]). Fig. 2 shows an overview of our learning framework. Our approach starts by a simple raw text representation involving term frequencies; see Sec 3.1. The text representation is then fed into a dimensionality reduction step followed by multi-part transformation to predict a visual classifier at the visual part level. The predicted classifier is applied on the part-based feature representations that are learnt through a deep Convolutional Neural Network (CNN). In the following subsections, we describe the text and visual part encoders, then define our

problem and the proposed approach on top of these encoders.

3.1. Text Encoder

Similar to [14, 26], text articles are first tokenized into words and the stop words are removed. Then, a simple Term Frequency-Inverse Document Frequency(TF-IDF) feature vector is extracted [41]. We denote the TF-IDF representation of a text article t by $\mathbf{t} \in \mathbb{R}^{d_T}$, where d_T is the number of terms in the TF-IDF text representation.

3.2. Visual Parts CNN Detector/Encoder (VPDE)

Detecting semantic parts facilitates modeling a representation that can be related to unstructured text terms at the part-level. It was shown in [51] that bird parts can be detected at precision of 93.40% vs 74.0% with earlier methods [27]. We adopt fast-RCNN framework [19] with VGG16 architecture [42] to detect seven small bird-parts using the small-part proposal method proposed in [51]. The seven parts in order are (1) head, (2) back, (3) belly, (4) breast, (5) leg, (6) wing and (7) tail; see Fig. 2. We denote the input image to the visual part encoder as x. First, the image x is processed through VGG16 convolutional layers. The proposed regions by [51] on x are then ROI pooled with a 3×3 grid. Then, they are then passed through an 8-way classifier (7 parts + background) and a bounding box regressor. Each part p is assigned to the region with the highest confidence of part p if that confidence is greater than a threshold (i.e. $1/7$). If the highest confidence of part p is less than the threshold, part p is considered as missing. The detected part regions are then passed to the visual encoder sub-network, which ROI(3×3) pools these regions and eventually encode each part into a 512 dimensional learning representation. When a part is missing, a region of all zeros is passed to the encoder-sub-network. We denote these part-learning representations of a bird image x as $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \cdots, \mathbf{x}^{(P)}$; see the flow from x to the part representation in Fig. 2 (top-part starting from the blue arrow at the top-left). We will detail later how the Visual Part Detector/Encoder (VPDE) network is trained. We denote the dimensionality of the part features as d_P, where $\mathbf{x}^{(p)} \in \mathbb{R}^{d_P} \forall p$ and $d_P = 512$ in our work.

3.3. Problem Definition

During training, the information comes from images and text descriptions of K seen classes. We denote the learning representations of the detected parts of N training examples as $\{\mathbf{X}^{(p)} \in \mathbb{R}^{d_P \times N}\}, p = 1 : P$, where P is the number of parts. We denote the text representation of K seen classes as $\mathbf{T} \in \mathbb{R}^{d_T \times K}$. We define $\mathbf{Y} \in \{0, 1\}^{N \times K}$ as the label matrix of each example in one-hot representation (i.e., each row in $\mathbf{Y}$ is a vector of zeros except at the corresponding class label index). At test time, the text features are given for $\hat{K}$ classes, where we need to assign the right label among them to each test image. Formally, the label assignment of an image x is defined as

$$k^* = \arg\max_k \sum_{p=1}^{P} z^{(p)}(\mathbf{t}_k)^\top \cdot \mathbf{x}^{(p)}, k = 1 : \hat{K} \quad (1)$$

where $\{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \cdots, \mathbf{x}^{(P)}\}$ is the part learned representation of image x, $\mathbf{t}_k$ is the text representation of class k, and $z^{(p)}(\mathbf{t})$ is a function that takes a text representation $\mathbf{t}$ and predicts a visual classifier weights for part p. In our work, we aim at jointly learning and regularizing $z^{(p)}(\cdot), \forall p \in 1 : P$ to encourage text terms to correlate with sparse set of parts.

3.4. Part Zero-Shot Classifier Prediction (PZSC)

Part visual classifier prediction functions are defined as

$$z^{(p)}(\mathbf{t}) = \mathbf{t}^\top \mathbf{W_t}^\top \mathbf{W}^p_\mathbf{x}, \forall p \in 1 : P \quad (2)$$

where $\mathbf{W_t} \in \mathbb{R}^{d \times d_T}$ is a dimensionality reduction matrix, which projects the text representation $\mathbf{t} \in \mathbb{R}^{d_T}$ into a latent space, $\mathbf{W}^p_\mathbf{x} \in \mathbb{R}^{d \times d_P}$ for each part p then regress the projected text representation into a classifier for part p; see Fig. 2 (bottom part starting from the blue arrow at the bottom-left). Hence, $z^{(p)}(\mathbf{t}) \forall p$ are mainly controlled by $\mathbf{W}^p_\mathbf{x}$ and $\mathbf{W_t}$ since $\mathbf{t}$ is the input. We will elaborate next on how $\mathbf{W_t}$ and $\mathbf{W}^p_\mathbf{x} \forall p$ are trained jointly.

3.5. Model Optimization and Training

An interesting research direction regularizes zero-shot learning by introducing different structures to the learning parameters (e.g., [39, 36]). In [39]. Minimizing the variance of the projections from image to attribute space and vice versa is the key to improving attribute-based zero-shot prediction. In [36], Qiao *et al.* used $l_{2,1}$ sparsity regularization, proposed in [32], to encourage sparsity on the text terms, and showed its capability to suppress noisy text terms and improve zero-shot classification from text. We got inspired by these regularization techniques to train our our framework in Fig. 2 with the following cost function:

$$\min_{\mathbf{W}^1_\mathbf{x}, \cdots, \mathbf{W}^P_\mathbf{x}, \mathbf{W_t}} ||(\sum_{p=1}^{P} \mathbf{X}^{(p)^\top} \mathbf{W}^{p\top}_\mathbf{x}) \mathbf{W_t} \mathbf{T} - \mathbf{Y}||^2_F + \lambda_1 \sum_{p=1}^{P} ||\mathbf{W}^{p\top}_\mathbf{x} \mathbf{W_t} \mathbf{T}||^2_F + \lambda_2 \sum_{p=1}^{P} ||\mathbf{W}^{p\top}_\mathbf{x} \mathbf{W_t}||_{2,1} \quad (3)$$

where $||\cdot||_F$ is the Frobenius norm. The first term in Eq. 3 encourages that for every image x_j, $\sum_{p=1}^{P} z^{(p)}(\mathbf{t}_k)^\top \cdot \mathbf{x}^{(p)}_j = \sum_{p=1}^{P} (\mathbf{t}^\top_k \mathbf{W_t}^\top \mathbf{W}^p_\mathbf{x})^\top \cdot \mathbf{x}^{(p)}_j$ to be equal to 1 if k is the ground truth class, 0 if other classes. This enables $z^{(p)}(\mathbf{t})$ to predict part classifiers for an arbitrary text $\mathbf{t}$ (i.e. high ($\rightarrow 1$) for the right class, low ($\rightarrow 0$) for others). The second term bounds the variance of the functions $\{z^{(p)}(\mathbf{t}) = \mathbf{t}^\top \mathbf{W_t}^\top \mathbf{W}^p_\mathbf{x} \ \forall p\}$. More importantly, the third term imposes structure on $\mathbf{W_t}$ and $\{\mathbf{W}^p_\mathbf{x} \forall p\}$, to encourage connecting every text term with sparse set of parts (i.e., every text term attends to as few parts as possible). The third term $\sum_{p=1}^{P} ||\mathbf{W}^{p\top}_\mathbf{x} \mathbf{W_t}||_{2,1}$ is defined

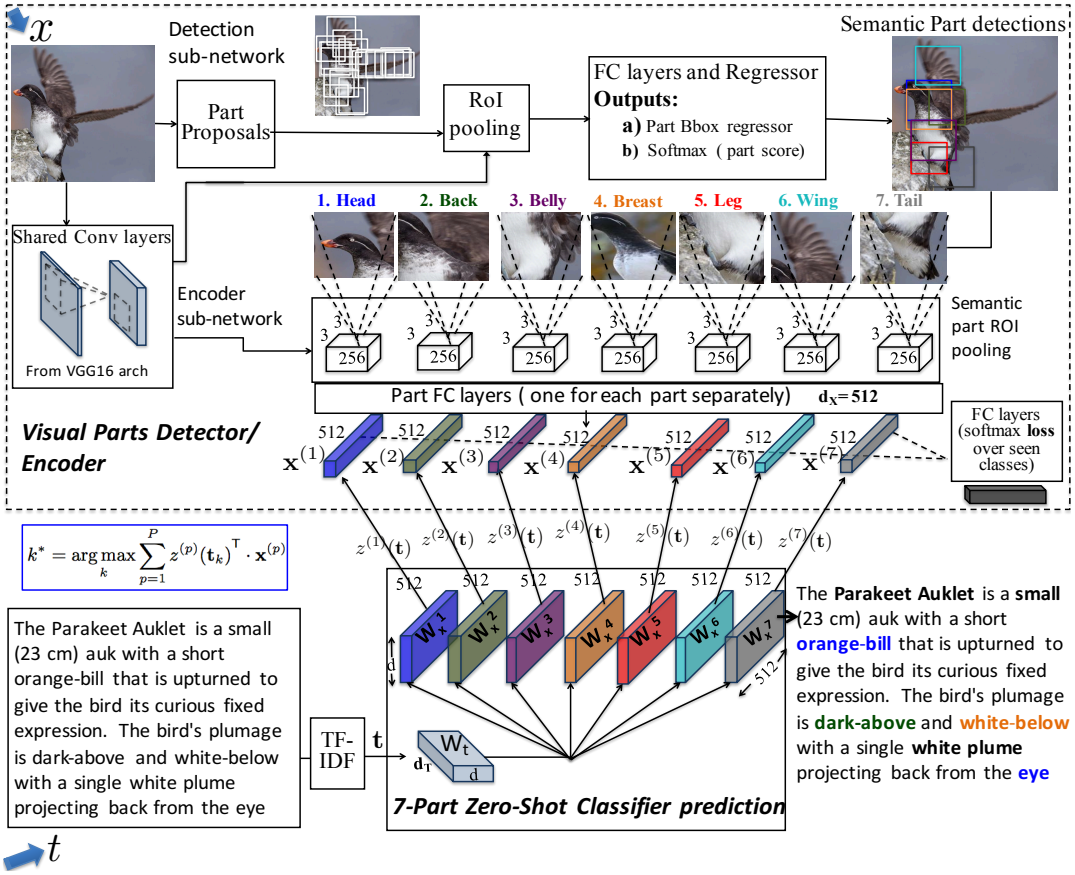

Figure 2. Our approach (best seen in color): On the bottom is the core of our approach where the input is a pure text description and produces classifier through a dimensionality reduction transformation W_t following by part projections $W_x^p, p = 1 : P$, where P is the number of parts. The produced P classifiers are then applied on the part learning representation produced through detected parts from the top visual CNN. RoI refers to Region of Interest Pooling [19].FC refers to Fully connected layers. VGG conv layers refer to the first five convolutional layers in VGGNet-16 [42]

as $\sum_{p=1}^{P}\sum_{i=1}^{d_T}||\mathbf{W}_\mathbf{x}^{p\top}\mathbf{w}_\mathbf{t}^i||_2$, $\mathbf{w}_\mathbf{t}^i$ is the i^{th} column in $\mathbf{W_t}$ matrix that corresponds to the i^{th} text term, $\mathbf{W}_\mathbf{x}^{p\top}\mathbf{w}_\mathbf{t}^i \in \mathbb{R}^{d_X}$ are the weights that connect the p^{th} part to i^{th} text term. Hence, the third term encourages group sparsity over the parameter groups that connect every text term i to every part p (i.e. $\mathbf{W}_\mathbf{x}^{p\top}\mathbf{w}_\mathbf{t}^i$), which encourages terms to be connected to parts sparsely.

Optimization: The parameters of our model include part detection sub-network parameters and part representation sub-network parameters for Visual Part Detector/Encoder (VPDE) network, and $\{\mathbf{W}_\mathbf{x}^p, p = 1 : P\}$, $\mathbf{W_t}$ for the part zero-shot classifier predictor (PZSC) network. The VPDE network is trained by alternate optimization over the detector and the representation sub-networks with the training images. The detector sub-network is optimized through softmax loss over 8 outputs (7 parts and background) and bounding box regression to predict the final box for each detected part. The representation sub-network is optimized over by softmax loss over the seen/training classes. The convolutional layers are shared between the detection and representation sub-networks (VGG16 conv layers in our work); see Fig. 2(top-part) and supplementary for architecture details. After training VPDE network, we solve the objective function in Eq. 3 to train the Part Zero-Shot Classifier predictor.

The cost function in Eq. 3 is convex if optimized for either $\mathbf{W_t}$ or $\{\mathbf{W}_\mathbf{x}^p, p = 1 : P\}$ individually but not convex on both. Hence, we solve Eq. 3 by an alternate optimization, where we fix $\mathbf{W_t}$ and solve for $\{\mathbf{W}_\mathbf{x}^p, p = 1 : P\}$, then fix $\{\mathbf{W}_\mathbf{x}^p, p = 1 : P\}$ and solve for $\mathbf{W_t}$.

Solving for $\mathbf{W_t}$: Following the efficient $l_{2,1}$ group sparsity optimization method in [32], the solution to this sub-problem could be efficiently achieved by sequentially solving to following problem until convergence.

$$\min_{\mathbf{W_t}, \{\mathbf{D}_l^p, \forall p\}} ||(\sum_{p=1}^{P}\mathbf{X}^{(p)\top}\mathbf{W}_\mathbf{x}^{p\top})\mathbf{W_t}\mathbf{T} - \mathbf{Y}||_F^2 + \lambda_1 \sum_{p=1}^{P}||\mathbf{W}_\mathbf{x}^{p\top}\mathbf{W_t}\mathbf{T}||_F^2 + \lambda_2 \sum_{p=1}^{P}\mathrm{Tr}\left(\mathbf{W}_\mathbf{x}^{p\top}\mathbf{W_t}\mathbf{D}_l^p\mathbf{W_t}^\top\mathbf{W}_\mathbf{x}^p\right) \tag{4}$$

where $\mathbf{D}_l^p$ is a diagonal matrix with the i-th diagonal element is $1/(2||\mathbf{W}_\mathbf{x}^p(\mathbf{w}_\mathbf{z}^i)^{(l-1)}||_2)^2$ at the the l-th iteration, where $(\mathbf{w}_\mathbf{z}^i)^{(l-1)}$ is the i-th column of $\mathbf{W_t}$ solution at iteration $l-1$.

Algorithm 1: Alternate Optimization to solve Eq. 3

Input : $\mathbf{T}, \mathbf{Y}, \mathbf{X}^{(1)}, \cdots \mathbf{X}^{(p)}$
Output: $\mathbf{W_t}, \mathbf{W}_\mathbf{x}^1, \cdots, \mathbf{W}_\mathbf{x}^P$

Initialize $\mathbf{W_t}$ and $\mathbf{W}_\mathbf{x}^1, \cdots, \mathbf{W}_\mathbf{x}^P$ with standard Gaussian distribution.
Initialize $\mathbf{Wt\_turn} = false$
for l=$1 \cdots L$ **do**
  Update $\mathbf{D}_l^{(p)}$ $\forall p$
  if *(*$\mathbf{Wt\_turn} = true$*)* **then**
    Find $\mathbf{W_t}$ with Eq. 4 by Quassi-Newton BFGS ;
  else
    Find $\{\mathbf{W}_\mathbf{x}^p\}$ with Eq. 5 by Quassi-Newton BFGS ;
  $\mathbf{Wt\_turn} = \mathbf{not}\ \mathbf{Wt\_turn}$
  if *Converges* **then**
    Break
  end
end

We realized that it is hard to find a closed-form solution to Eq. 4 or even reduce it to the Sylvester Equation [8]. Hence, we solve Eq. 4 by Quasi-Newton with Limited Memory BFGS Updating (i.e., gradient-based optimization). The derived gradients for Eq. 4 sub-problem are attached in the supplementary materials.

Solving for $\mathbf{W}_\mathbf{x}^p$: In this step, we solve the following sub-problem.

$$\min_{\{\mathbf{D}_l^p, \mathbf{W}_\mathbf{x}^p, \forall p\}} ||(\sum_{p=1}^{P} \mathbf{X}^{(p)^\top} \mathbf{W}_\mathbf{x}^{p\top}) \mathbf{W_t T} - \mathbf{Y}||_F^2 + \lambda_1 \sum_{p=1}^{P} ||\mathbf{W}_\mathbf{x}^{p\top} \mathbf{W_t T}||_F^2 + \lambda_2 \sum_{p=1}^{P} \mathrm{Tr}\left(\mathbf{W}_\mathbf{x}^{p\top} \mathbf{W_t} \mathbf{D}_l^p \mathbf{W_t}^\top \mathbf{W}_\mathbf{x}^p\right) \tag{5}$$

where $\mathbf{D}_l^p$ is a diagonal matrix with the i-th diagonal element is $1/(2||(\mathbf{W}_\mathbf{x}^p)^{(l-1)} \mathbf{w}_\mathbf{z}^i||_2)^2$ at the l-th iteration, where $(\mathbf{W}_\mathbf{x}^p)^{(l-1)}$ is the solution of $\mathbf{W}_\mathbf{x}^p$ for part p at iteration $l-1$. Similar to Eq. 4, we solve Eq. 5 by Quasi-Newton with BFGS Updating. The derived gradients for Eq. 5 sub-problem are attached in the supplementary materials. Algorithm 1 shows the overall optimization process that solves $\mathbf{W_t}$ and $\mathbf{W}_\mathbf{x}^1, \cdots, \mathbf{W}_\mathbf{x}^P$ jointly.

4. Experiments

4.1. Experiment setting

Datasets: We compare the proposed method with state-of-the-art approaches on two datasets: CUB2011 [48] and NABirds [44]. Both are bird datasets for fine-grained classification. Important parts of the bird in each image are annotated with locations by experts. CUB2011 dataset contains 200 categories of bird species with a total of 11,788 images. Compared with CUB2011, NABirds is a larger dataset of birds with 1011 classes and 48562 images. It constructs a hierarchy of bird classes, including 555 leaf nodes and 456 parent nodes, starting from the root class "bird". Only leaf nodes are associated with images, and the images for parent class can be collected by merging all images of its children nodes. In practice, we found some pairs of classes merely differ in gender. For example, the parent node "American Kestrel" are divided to "American Kestrel (Female, immature)" and "American Kestrel (Adult male)". Since we cannot find the Wikipedia articles for this subtle division of classes, we merged such pairs of classes to their parent. After such processing, we finally have 404 classes, each one is associated with a set of images, as well as the class description from Wikipedia. We collected the raw textual sources from English-language Wikipedia-v01.02.2016. We manually verified all the articles and augmented classes with limited descriptions from the all-about-birds website [2]. We plan to release this data and the NABird benchmarks that we set up.

Two split setting: To split the dataset to training/testing set, we have designed two kinds of splitting schemes, in terms of how close the seen classes are to the unseen classes: Super-Category-Shared splitting (SCS), Super-Category-Exclusive splitting(SCE). In the dataset, some classes often are the further division of one category. For example, both "Black footed Albatross" and "Laysan Albatross" belong to the category "Albatross" in CUB2011, and both "Cooper's Hawk" and "Harris's Hawk" are under the category "Hawks" in NABirds. For SCS, unseen classes are deliberately picked in the condition that there exists seen classes with the same Super-Category. In this scheme, the relevance between seen classes and unseen classes is very high. On the contrary, in SCE, all classes under the same category as unseen classes would either belong to the seen or the unseen classes. For instance, if "Black Footed Albatross" is an unseen class then all other albatrosses are unseen classes as well and so no albatrosses are seen during training. It is not hard to see that the relevance between seen and unseen classes is minimal in the SCE-split. Intuitively, SCE-split is much harder compared to SCS-split.

These strategies for zero-shot splits were used on CUBirds dataset in the literature but in different works and were not compared to each other. *For SCS-split on CUB2011*, we use the same splitting to [3, 36], where 150 classes for training and 50 classes for testing. *For SCE-split on CUB2011*, we use the same splitting to [14], where the first 80% classes are considered as seen classes and used for training. To design these two splitting schemes in the NABirds, we first check the class hierarchy. There exist 22 children nodes under the root category (bird) in the hierarchy. We found that the number of descendants under the 22nd children (Perching Birds) are much greater than the average descendants of the remaining 21 classes (205 vs.10). To eliminate this imbalance, we further divide this category to its children. With the combination of 29 children of this category and

other 21 children of the root, we ended up with 50 super categories (21+29). For SCS-split, we randomly pick 20% of descendant classes under each super categories as unseen classes. For SCE-split, we randomly pick 20% of super categories and consider all their-descendant classes as unseen are considered the seen classes. For both splits, there are totally 323 training (seen) classes and 81 testing (unseen) classes, respectively. For ease of presentation, we sometimes refer to the SCS-split as the easy-split and to SCE-split as the hard-split.

Textual Representation: We extract the text representation according to the scheme described in Section 3.1. The dimensionality of TF-IDF feature for CUB2011 and NABirds are 11083 and 13585, respectively.

Image representation: As described in Section 3.2, the part regions are first detected and then passed to the VPDE network. 512-dimensional feature vector is extracted for each semantic part. For CUB2011 dataset, we only use seven semantic parts to train the VPDE network; illustrated in Fig. 2. For NABird dataset, we used only six visual parts with the "leg" part removed, since there is no annotations for the "leg" part in the NABirds dataset.

4.2. Performance evaluation

Baselines and Competing Methods: The performance of our approach is compared to six state-of-the-art algorithms: SJE [6], MCZSL [3], ZSLNS [36], ESZSL [39], WAC [14]. The source code of ESZSL and ZSLNS are available online, and we get the code of WAC [14, 13] from its author. For MCZSL and SJE, since their source codes are not available, we directly copy the highest scores for non-attribute settings reported in [3, 6]. *Image-sentence baseline [46]:* Additionally, we used a state of the art Model [46] for image-sentence similarity by breaking down each text document into sentences and considering it as a positive sentence for all images in the corresponding class. Then we measure the similarities between an image to class by averaging its similarity to all sentences in that class. Images were encoded using VGGNet [42] and sentences were encoded by an RNN with GRU activations [12]. The purpose of this experiment is to study how RNN representation of the sentences perform in our setting with noisy text descriptions.

We first compare our approach with MCZSL, which is among the best performing state-of-art methods. Both our approach and MCZSL utilizes part annotations provided by the CUB2011 datasets.However, in contrast to MCZSL, which directly uses part annotations to extract image feature in the test phase, our approach is merely based on the detected semantic parts during both training and testing. Less accurate detection of semantic parts will surely degrade the accuracy for the final zero-shot classification. In order to make a fair comparison with MCZSL, we also report our result using the ground-truth annotations of semantic parts at test-time. The results of our approach based on the detected parts and ground-truth parts are denoted by "Ours-DET" and "Ours-ATN", respectively. In Table 1, we compared to the same benchmark reported in [3], which is the SCS-split on CUBirds 2011 dataset. The results show that our performance is 9% better than [3] (43.6% vs 34.7%) although we only used a simple TF-IDF text representation compared to multiple cues used in MCZSL like text, WordNet and word2vec. Note also that the 34.7% achieved by [3] used 19 part annotations during training and testing (the whole image, head, body, full object, and 15 part locations annotated), while we only used 7 parts to achieve the 43.6%. Table 1 also shows that our method still perform 2.5% better even when using the detected parts at test time (37.2% Ours-DET vs 34.7% MCSZSL using ground truth annotations). In all the following experiments, we only used our approach with the detected parts (i.e. "Ours-DET").

methods	Accuracy
MCZSL [3](BoW)	26.0
MCZSL [3](word2vec)	32.1
MCZSL [3](Comb)	34.7
Ours-DET	37.2
Ours-ATN	**43.6**

Table 1. Performance comparison with the accuracy (%) on **CUB2011** Dataset. In [3], the approach is evaluated with different textual representation: BoW, word2vec, and their combination.

Zero-shot Top-1 Accuracy. For standard zero-shot image classification, we calculate the mean Top-1 accuracy obtained on unseen classes. We performed comprehensive experiments on both SCS-(easy) and SCS-(hard) splits on both CUBirds and NABirds. Note that some of these methods were applied on attributes prediction (e.g., ZSLNS [36], SynC [10], ESZSL [39]) or image-sentence similarity (e.g.,Order Embedding [46]). We used the publicly available code of these methods and other text-based methods like (ZSLNS [36], WAC [14], WAC-kernel [13]) to apply them on our setting. Note that the conventional split setting for zero-shot learning is Super-Category Shared splitting (i.e. SCS-(easy) split). We think evaluating the performance on both the SCS-(easy) and the SCE-(hard) splits are complementary and hence we report the performance on both of them. In Table 2, we show the comparisons between our method to all the baselines on the CUB2011 easy and hard benchmarks, where method outperforms all the baselines by a noticeable margin on both the easy and the hard benchmarks. Note that the image-sentence similarity baseline (i.e. Order Embedding [46]) is among the least-performing methods. We think the reason is the level of noise which is addressed by the other methods by regularizing the text information at the term level, while the representation unit in [46] is the whole sentence. Similarly, Table 3 shows the results on NABirds easy and hard benchmarks, where the performance of our approach is also superior over the competing methods. It is worth mentioning that the WAC-method is not

scalable since the its training parameters depend on the number of image-class pair. We trained it for 6 days on 64GB RAM machine and report the results of the latest snapshot in Table 3.

methods	SCS(Easy)	SCE(Hard)
WAC-Linear [14]	27.0	5.0
WAC-Kernel [13]	33.5	7.7
ESZSL [39]	28.5	7.4
SJE [6]	29.9	–
ZSLNS [36]	29.1	7.3
SynC$_{fast}$ [10]	28.0	8.6
SynC$_{OVO}$ [10]	12.5	5.9
Order Embedding [46]	17.3	5.9
Ours-DET	**37.2**	**9.7**

Table 2. Top-1 accuracy (%) on **CUB2011** Dataset in two different split settings. Note that some of these methods are attribute-based methods but applicable in our setting by replacing attribute vectors with text features.

methods	SCS(Easy)	SCE(Hard)
WAC-Kernel [13]	11.4	6.0
ESZSL [39]	24.3	6.3
ZSLNS [36]	24.5	6.8
SynC$_{fast}$ [10]	18.4	3.8
Ours-DET	**30.3**	**8.1**

Table 3. Top-1 accuracy (%) on **NABird** Dataset splits.

Generalized Zero-Shot Learning Performance. The conventional zero-shot learning that we discussed earlier classifies test examples into unseen classes without considering the seen classes in test phase. Because the seen classes are often the most common, it is hardly realistic to assume that we will never encounter them during the test phase [11]. To get rid of such an assumption, Chao *et al*. [11] recently proposed a more general metric for generalized zero-shot learning (GZSL). We here briefly review how it generally measures the capability of recognizing not only unseen data, but also seen data. Let $\mathcal{S}, \mathcal{U}$ denote the label spaces of seen classes, unseen classes; $\mathcal{T} = \mathcal{S} \cup \mathcal{U}$, the joint label space. $A_{\mathcal{U}\to\mathcal{T}}$ and $A_{\mathcal{S}\to\mathcal{T}}$ are the accuracies of classifying seen data and unseen data into joint label space. The labels are computed using the Eq. 6:

$$y = \arg\max_{c\in\mathcal{T}} f(\mathbf{x}) - \lambda I[c \in \mathcal{S}] \tag{6}$$

where $I[.] \in \{0, 1\}$ indicates whether c is a seen class and λ is the penalty factor. $\mathbf{x}$ is set to seen data or unseen data to calculate $A_{\mathcal{U}\to\mathcal{T}}$ and $A_{\mathcal{S}\to\mathcal{T}}$, respectively. As λ increases or decreases, data are encouraged to be classified to unseen classes or seen classes, respectively. In the cases where λ is extremely large or small, all data will assigned with unseen class label or seen class label, respectively. Therefore, we can generate a series of pairs of classification accuracies ($\langle A_{\mathcal{U}\to\mathcal{T}}, A_{\mathcal{S}\to\mathcal{T}}\rangle$) by tuning values of λ. Considering these pairs as points with $A_{\mathcal{U}\to\mathcal{T}}$ as x-axis and $A_{\mathcal{S}\to\mathcal{T}}$ as y-axis, we can draw the Seen-Unseen accuracy Curve(SUC). The Area Under SUC (AUSUC), as a widely-used measure of curves, can well assess the performance of an classifier in balance of the conflicting $A_{\mathcal{U}\to\mathcal{T}}$ and $A_{\mathcal{S}\to\mathcal{T}}$) measurements.

The Seen-Unseen accuracy Curve of our method and other state-of-the-art approaches are shown in Fig. 3. The performance of our work is superior over all other methods in term of the AUSUC score. Although WAC_linear apparently achieves a high performance on seen classes, its poor performance of classifying unseen classes indicates that it doesn't learn much knowledge that can be effectively transferred to unseen classes. On the contrary, ZSLNS has a relatively good accuracy $A_{\mathcal{U}\to\mathcal{T}}$, but its lower $A_{\mathcal{S}\to\mathcal{T}}$ compared with other methods indicates that the success of unseen classes' classification may come from the overweighted regularizers. Our method remarkably outperforms other methods in term of both the classification of unseen classes, and also achieves a relative high accuracy in recognition of seen classes. The curves in Fig. 3 demonstrate our method's capability of balancing the classification of unseen classes and seen classes (0.304 AUSUC for Ours-DET compared to 0.239 for the best performing baseline). We also demonstrated the effectiveness of our performance on NABirds dataset in Fig. 4 (0.126 AUSUC for Ours-DET compared to 0.093 for the best performing baseline). In addition to these GZSL results on SCS-splits, we also report the Seen/Unseen curves on the SCE-splits in the supplementary due to space.

Model Analysis and Qualitative Examples. We also analyzed the the connections between the terms and parts in the learnt parameters, which is $\mathbf{W}_{\mathbf{x}}^{p\top}\mathbf{w}_{\mathbf{t}}^{i}$ for the connection between term i and part p on CUBirds dataset (SCS-split). Fig. 6 shows the l_2 norm of $\mathbf{W}_{\mathbf{x}}^{p\top}\mathbf{w}_{\mathbf{t}}^{i}$ for each part separately and only on the top 30 terms for each part sorted by $\|\mathbf{W}_{\mathbf{x}}^{p\top}\mathbf{w}_{\mathbf{t}}^{i}\|_2$. Fig. 8 shows the percentage of overlap between these terms for every pair of parts, which shows that every part focus on its relevant concepts yet there is still a shared portion that includes shared concepts like color and texture. In Fig. 6, we show the the summation of these connections for every part and compare between "Ours-DET" and "Ours-ATN" to analyze the effect of detecting the parts versus using part annotations. We observe that more concepts/terms are discovered and connected to head for "Ours-ATN", while more concepts are learnt for "breast" for "Ours-DET". This is also consistent with the Top-1 accuracy if each part is individually used for recognition; see the Top-1 Acc for each part separately in Fig. 6 (right). This observation shows if we have a perfect detector, head will be one of the most important part to be connected to terms which is intuitive. We also observed the same conclusion on both SCS and SCE splits on NABirds and SCE on CUBirds; see additional analysis figures for these splits in the supplementary. We further demonstrate these part-to-term connectivity by some qualitative examples in Fig. 7. For each bird, the top related term for each part is printed based on ranking

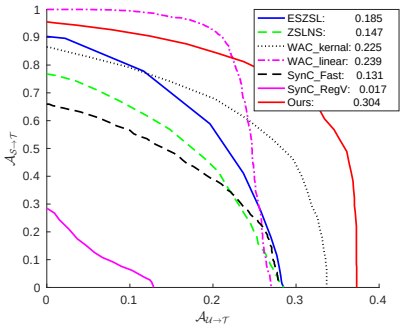

Figure 3. CUBirds Seen-Unseen accuracy Curve (SCE split)

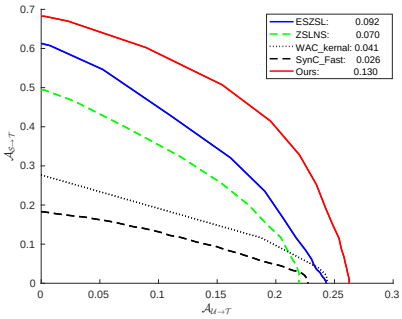

Figure 4. NABirds Seen-Unseen accuracy Curve (SCS split)

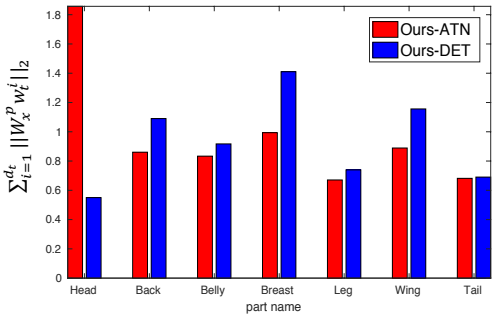

Figure 5. Connection to Text Terms

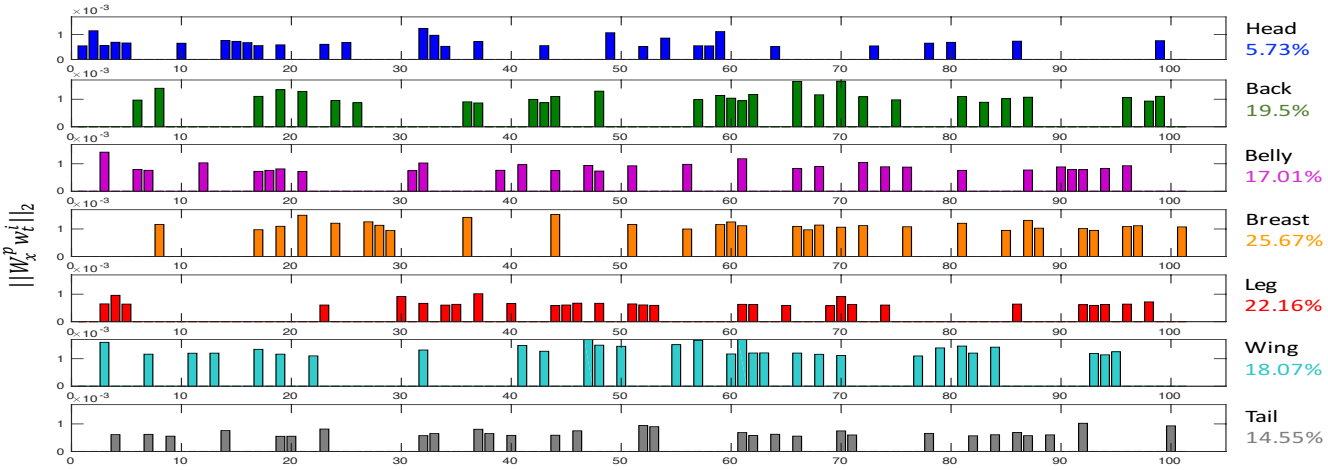

Figure 6. Connection to Text Terms (CU Birds dataset–SCS Split with with 37.2% Top1-Acc). On the right, Top1-Acc is shown per part

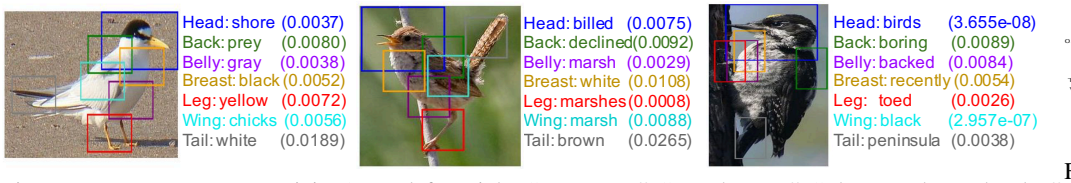

Figure 7. Part-to-Term connectivity (From left to right: "Least Tern", "Marsh Wren", "Three-toed Woodpecker" from CUBirds-SCS split)

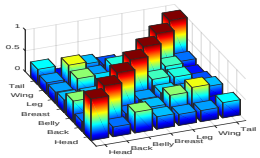

Figure 8. Top-30 terms Overlap between every two parts(CUBirds-SCS)

the terms by $\mathbf{x}^{(p)}\mathbf{W}_{\mathbf{x}}^{p\top}\mathbf{w}_{\mathbf{z}}^{i}\mathbf{t}_{k}^{i}$, where $\mathbf{t}_{k}^{i}$ is the i^{th} dimension of the text representation of the predicted class k (i.e., only the text terms that exist in the text description of class k are considered). The figure shows the capability of our method to ground concepts to its location in the image. In the right example, like "toes" is strongly connected to leg– the connection strength is shown between parenthesis. In the middle example, "billed" concept is connected to head, "white" is connected to the breast, and "brown" is connected to the tail. In the left example, "yellow" is connected to leg.

5. Conclusion

We developed a novel method for zeros-shot fine-grained recognition with a capability to connect terms to bird parts without requiring part-term annotations. Our learning framework is composed of Visual Part Detector/ Encoder (VPDE-net) that detects bird parts and learnt its representation, and part-based Zeros-Shot Classifier Predictor network (PZSC-net), that predict visual classifier function for every part. These part classifier prediction functions are jointly learnt to encourage text terms to be connected to the sparse set of parts, which help suppress the noise in the text and enable connecting terms to relevant parts. Our method significantly outperforms existing methods on two existing benchmarks: CUB2011 dataset and large-scale benchmarks that we created on NABirds dataset. We also performed an analysis on the part-to-text connection weights that our model learns and we discussed interesting findings.

Acknowledgment. This work was supported NSF-USA award #1409683.

References

[1] Our implementation: ZSL PP. https://github.com/EthanZhu90/ZSL_PP, 2017. 1

[2] T. C. L. B. Academy. All About Birds. info.allaboutbirds.org, 2016. [Online; accessed 19-June-2016]. 2, 5

[3] Z. Akata, M. Malinowski, M. Fritz, and B. Schiele. Multi-cue zero-shot learning with strong supervision. *arXiv preprint arXiv:1603.08754*, 2016. 2, 5, 6

[4] Z. Akata, F. Perronnin, Z. Harchaoui, and C. Schmid. Label-embedding for attribute-based classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 819–826, 2013. 1, 2

[5] Z. Akata, F. Perronnin, Z. Harchaoui, and C. Schmid. Label-embedding for image classification. *IEEE transactions on pattern analysis and machine intelligence*, 38(7):1425–1438, 2016. 1, 2

[6] Z. Akata, S. Reed, D. Walter, H. Lee, and B. Schiele. Evaluation of output embeddings for fine-grained image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2927–2936, 2015. 1, 2, 6, 7

[7] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. Lawrence Zitnick, and D. Parikh. Vqa: Visual question answering. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2425–2433, 2015. 2

[8] R. H. Bartels and G. W. Stewart. Solution of the matrix equation ax + xb = c [f4]. *Commun. ACM*, 15(9):820–826, Sept. 1972. 5

[9] T. L. Berg, A. C. Berg, and J. Shih. Automatic attribute discovery and characterization from noisy web data. In *European Conference on Computer Vision*, pages 663–676. Springer, 2010. 2

[10] S. Changpinyo, W.-L. Chao, B. Gong, and F. Sha. Synthesized classifiers for zero-shot learning. *arXiv preprint arXiv:1603.00550*, 2016. 1, 6, 7

[11] W.-L. Chao, S. Changpinyo, B. Gong, and F. Sha. *An Empirical Study and Analysis of Generalized Zero-Shot Learning for Object Recognition in the Wild*. 2016. 7

[12] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. In *Conference on Empirical Methods in Natural Language Processing (EMNLP 2014)*, 2014. 6

[13] M. Elhoseiny, A. Elgammal, and B. Saleh. Write a classifier: Predicting visual classifiers from unstructured text descriptions. *arXiv preprint arXiv:1601.00025*, 2015. 6, 7

[14] M. Elhoseiny, B. Saleh, and A. Elgammal. Write a classifier: Zero-shot learning using purely textual descriptions. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2584–2591, 2013. 1, 2, 3, 5, 6, 7

[15] H. Fang, S. Gupta, F. Iandola, R. K. Srivastava, L. Deng, P. Dollár, J. Gao, X. He, M. Mitchell, J. C. Platt, et al. From captions to visual concepts and back. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1473–1482, 2015. 2

[16] A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth. Describing objects by their attributes. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1778–1785. IEEE, 2009. 1, 2

[17] Y. Fu, T. M. Hospedales, T. Xiang, Z. Fu, and S. Gong. Transductive multi-view embedding for zero-shot recognition and annotation. In *European Conference on Computer Vision*, pages 584–599. Springer, 2014. 2

[18] A. Fukui, D. H. Park, D. Yang, A. Rohrbach, T. Darrell, and M. Rohrbach. Multimodal compact bilinear pooling for visual question answering and visual grounding. *arXiv:1606.01847*, 2016. 2

[19] R. Girshick. Fast r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1440–1448, 2015. 3, 4

[20] S. Huang, M. Elhoseiny, A. Elgammal, and D. Yang. Learning hypergraph-regularized attribute predictors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 409–417, 2015. 2

[21] S. J. Hwang and L. Sigal. A unified semantic embedding: Relating taxonomies and attributes. In *Advances in Neural Information Processing Systems*, pages 271–279, 2014. 2

[22] A. Karpathy and L. Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3128–3137, 2015. 2

[23] R. Kiros, R. Salakhutdinov, and R. S. Zemel. Unifying visual-semantic embeddings with multimodal neural language models. *TACL*, 2015. 2

[24] 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, 2009. CVPR 2009. IEEE Conference on*, pages 951–958. IEEE, 2009. 1, 2

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

[26] J. Lei Ba, K. Swersky, S. Fidler, et al. Predicting deep zero-shot convolutional neural networks using textual descriptions. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4247–4255, 2015. 1, 2, 3

[27] D. Lin, X. Shen, C. Lu, and J. Jia. Deep lac: Deep localization, alignment and classification for fine-grained recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1666–1674, 2015. 3

[28] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision*, pages 740–755. Springer, 2014. 2

[29] T. Mensink, E. Gavves, and C. G. Snoek. Costa: Co-occurrence statistics for zero-shot classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2441–2448, 2014. 2

[30] 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*, pages 3111–3119, 2013. 2

[31] G. A. Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995. 2

[32] F. Nie, H. Huang, X. Cai, and C. H. Ding. Efficient and robust feature selection via joint 2, 1-norms minimization. In *Advances in neural information processing systems*, pages 1813–1821, 2010. 3, 4

[33] D. Parikh and K. Grauman. Relative attributes. In *2011 International Conference on Computer Vision*, pages 503–510. IEEE, 2011. 2

[34] J. Pennington, R. Socher, and C. D. Manning. Glove: Global vectors for word representation. In *EMNLP*, volume 14, pages 1532–43, 2014. 2

[35] B. A. Plummer, L. Wang, C. M. Cervantes, J. C. Caicedo, J. Hockenmaier, and S. Lazebnik. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2641–2649, 2015. 2

[36] 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), 2016 IEEE Conference on, year=2016*. 1, 2, 3, 5, 6, 7

[37] M. Rohrbach, M. Stark, G. Szarvas, I. Gurevych, and B. Schiele. What helps where–and why? semantic relatedness for knowledge transfer. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 910–917. IEEE, 2010. 2

[38] M. Rohrbach, M. Stark, G. Szarvas, and B. Schiele. Combining language sources and robust semantic relatedness for attribute-based knowledge transfer. In *European Conference on Computer Vision*, pages 15–28. Springer, 2010. 2

[39] B. Romera-Paredes and P. Torr. An embarrassingly simple approach to zero-shot learning. In *Proceedings of The 32nd International Conference on Machine Learning*, pages 2152–2161, 2015. 1, 2, 3, 6, 7

[40] R. Salakhutdinov, A. Torralba, and J. Tenenbaum. Learning to share visual appearance for multiclass object detection. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1481–1488. IEEE, 2011. 1

[41] G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. *Information processing & management*, 24(5):513–523, 1988. 3

[42] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. 3, 4, 6

[43] G. Van Horn, S. Branson, R. Farrell, S. Haber, J. Barry, P. Ipeirotis, P. Perona, and S. Belongie. Building a bird recognition app and large scale dataset with citizen scientists: The fine print in fine-grained dataset collection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 595–604, 2015. 2

[44] G. Van Horn, S. Branson, R. Farrell, S. Haber, J. Barry, P. Ipeirotis, P. Perona, and S. Belongie. Building a bird recognition app and large scale dataset with citizen scientists: The fine print in fine-grained dataset collection. In *Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, 2015. 5

[45] I. Vendrov, R. Kiros, S. Fidler, and R. Urtasun. Order-embeddings of images and language. *ICLR*, 2016. 2

[46] I. Vendrov, R. Kiros, S. Fidler, and R. Urtasun. Order-embeddings of images and language. In *ICLR*, 2016. 6, 7

[47] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan. Show and tell: A neural image caption generator. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3156–3164, 2015. 2

[48] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD Birds-200-2011 Dataset. Technical report, 2011. 1, 2, 5

[49] X. Wang and Q. Ji. A unified probabilistic approach modeling relationships between attributes and objects. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2120–2127, 2013. 1

[50] P. Young, A. Lai, M. Hodosh, and J. Hockenmaier. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the Association for Computational Linguistics*, 2:67–78, 2014. 2

[51] H. Zhang, T. Xu, M. Elhoseiny, X. Huang, S. Zhang, A. Elgammal, and D. Metaxas. Spda-cnn: Unifying semantic part detection and abstraction for fine-grained recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 3

[52] X. Zhu, D. Anguelov, and D. Ramanan. Capturing longtail distributions of object subcategories. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 915–922, 2014. 1