Using Attribute Embeddings for Fine-grained Zero-shot Image Classification

Danyi (Oli) Liu
Christ's College
dl567@cam.ac.uk

Abstract

This project[1] *investigated the use of attribute-based embeddings in performing the fine-grained zero-shot image classification task. Using attribute embeddings extracted from ResNet models fine-tuned for attribute prediction, which are of much lower dimension than the original ResNet feature representation, the accuracy on predicting novel classes unseen in training with a mapping-based zero-shot classifier was almost doubled from 17.59% to 32.69%.*

1. Introduction

Deep learning models have achieved significant success on supervised computer vision tasks. On the challenging task of bird species classification [14], vision transformers can obtain accuracies as high as 91.7% [4]. However, one of the persisting challenges faced by state-of-the-art image classification systems in real world deployment is the presence of novel classes unseen in training data. To tackle this problem, zero-shot learning (ZSL) is proposed as a learning paradigm for targeted investigation. ZSL methods endeavours to account for novel classes based on generalisation of knowledge contained in classes present in the training set. Throughout this work, ZSL is used to refer to transductive as opposed to inductive zero-shot learning where no side-information of novel classes are available.

Modern ZSL approaches fall into two main categories: classifier-based and instance-based methods [16]. Classifier-based methods focus on learning classifiers for unseen classes that leverage knowledge about the classes in the training set, while instance-based methods try to exploit the training set to synthesise features of unseen classes using generative models including Generative Adversarial Networks (GANs) [19] and likelihood-based methods [11]. However, features generated by GANs often lack diversity due to mode collapse while likelihood-based Variational Autoencoders suffer from posterior collapse, creating non-discriminative features. In contrast, classifier-based methods tend to be much easier to train as well as more scalable. Among classifier-based models, mapping-based methods have gained particular popularity. These methods aim at acquiring cross-modal mapping between the image and class embedding spaces so that the compatibility between the an image and a class — with the class represented by corresponding textual descriptions or attributes encoding — can be measured and ranked.

In this project, I have focused on improving the performance of mapping-based ZSL methods on fine-grained image classification by replacing holistic image embeddings with compositional attribute representations. Although early works in ZSL proposed two-stage models that exploit attributes to create associations between seen and unseen classes [15, 9], these models preceded the development of powerful discriminative convolutional neural networks (CNNs). Instead, they were either entirely generative or contain generative sub-modules, and yielded poor performances. More recently, attribute representations extracted with CNNs have proved helpful for fine-grained image classification [3]. In the meantime, the compositional relationship between classes and attributes have also been used in instance-based ZSL methods [7]. However, there remains the research gap of exploring the use of attribute neural embeddings to perform zero-shot fine-grained classification via mapping-based methods, which this project aims to explore with the Caltech-UCSD Birds-200-2011 dataset [14].

This report starts by introducing the related previous work in the three research topics pertaining to this work to provide motivation and delineate the research gap this work aims to fill. It goes on to explain the details of the specific models used in Section 3 and describes the implementation details in Section 4. Having presented the results in Section 5, the report concludes with a discussion of the contribution and implications of this study.

2. Related Work

Fine-grained Image Classification Fine-grained image classification is the task of distinguishing between sub-

[1] code available here

categories within a meta-category, *e.g.* birds, cars, aircrafts. The task poses challenges to modern deep learning models that have achieved superior performance on generic image classification tasks because there is small inter-class variations, which renders relatively large intra-class variations caused by differences in poses, scales, and backgrounds. While earlier approaches for fine-grained image classification (FGIC) relied on multi-stage mechanisms and dense part annotations e.g. [20, 8], other methods learnt region-based features in an end-to-end framework using techniques like attention [12]. Apart from these part-based models, end-to-end discriminative feature extractors have gained popularity over the years. A notable example is the bilinear CNNs [10] that use pooled outer products of two deep CNNs to featurise input images. For a detailed survey of fine-grained image analysis with deep learning, refer to [17].

Zero-shot Learning While Xian *et al*. [18] presented detailed evaluation and comparison of the mapping-based ZSL models, which served as a baseline for this project, a more general survey that encompasses a broader range of models was given in [16]. Mapping-based methods relates input embeddings, *i.e.* image features, and output embeddings, *i.e.* encoded descriptions or attributes, through a compatibility function. Mapping-based methods described in [18] differed in the choice of input/output embeddings as well as compatibility function. The model deployed in this project was based on the Structured Joint Embedding introduced in [1], although most of the other mapping-based methods can also benefit from the formulation of input embeddings proposed here. While Akata *et al*. [1] placed significant emphasis on comparing and combining different output embeddings, I ignored words embedding and fixed output embeddings to attribute encodings in this work so as to focus on the creation of input embeddings. Finally, generalised ZSL was proposed as a setting that more closely reflects the real-world scenario where there are test samples belonging to both seen and unseen classes [18].

Attribute Learning The motivation for attribute learning is that categories can be viewed as a composition of visual attributes [2]. Thus, attribute learning acts as a compelling way to incorporate human-understandable semantic knowledge to supplement visual samples. Particularly relevant to this project is the use of attribute learning in ZSL. Wang *et al*. [15] introduced a generative model for each individual attribute while Lampert *et al*. [9] proposed discriminative models that predict probability of each attribute. The class of the image can then be determined using Bayes' rule. Finally, Han *et al*. [3] proposed an attention model that learn local attribute representations and global category representations in an end-to-end manner. Via the attention mechanism, the attribute information helped to select key features for fine-grained classification and yielded superior empirical results. Although Han *et al*. [3] did not explore or report performance that can be achieved by attribute embeddings alone but rather exploited the synergy of local attribute embeddings and global image embeddings, their results provided evidence for the efficacy of attribute embeddings in learning discriminative features in the setting of fine-grained image classification.

3. Method

3.1. Structured Joint Embedding

The general mapping-based ZSL framework was formalised in [1] as follows:

Given the training set

$$\mathcal{S} = \{(x_n, y_n), n = 1, ..., N\}, \text{ where } x_n \in \mathcal{X} \text{ and } y_n \in \mathcal{Y}$$

SJE learns the mapping $f : \mathcal{X} \rightarrow \mathcal{Y}$ by minimising the empirical loss

$$\frac{1}{N} \sum_{n=1}^{N} \Delta(y_n, f(x_n))$$

where $\Delta : \mathcal{Y} \times \mathcal{Y} \rightarrow \mathbb{R}$ is the cost of the mismatch between the label y_n and the prediction $f(x_n)$.

The mapping f is commonly formulated as a function returning the class label that maximises a compatibility function. SJE uses a linear compatibility measure which can be represented as:

$$F(x, y; W) = \theta(x)^T W \phi(y)$$

where $\theta(x)$ and $\phi(y)$ are input and output embeddings with dimendions d_{in} and d_{out} respectively. The compatability measur F is parameterised by $W : d_{\text{in}} \times d_{\text{out}}$.

Inspired by structured SVMs [13], SJE uses the following objective during training

$$\frac{1}{N} \sum_{n=1}^{N} \max_{y \in \mathcal{Y}} \{0, l(x_n, y_n, y)\}$$

$$l(x_n, y_n, y) = \Delta(y_n, y) + \theta(x)^T W \phi(y) - \theta(x)^T W \phi(y_n)$$

$$\Delta(y_n, y) = 1 - \mathbb{I}(y_n, y)$$

Finally, SJE uses pretrained embeddings (θ, ϕ) and applys Stochastic Gradient Descent (SGD) for optimisation, *i.e.*, W is updated as follows at each training step:

$$W^{(t+1)} = W^{(t)} + \eta_t \theta(x_n)[\phi(y_n) - \phi(y)]^T$$

3.2. Attribute Embedding Extractor

To extract attribute embeddings, I trained one attribute prediction CNN model for each attribute by finetuning ResNet-101 [5]. To obtain attribute embedding, I simply removed the final classification layer of the trained model and took the output of the last hidden layer.

ResNets are a class of powerful CNN models that make use of residual connections that skip hidden layers so as to avoid vanishing gradients, a problem that once limited the depth of neural network models. In He *et al*. [5], the

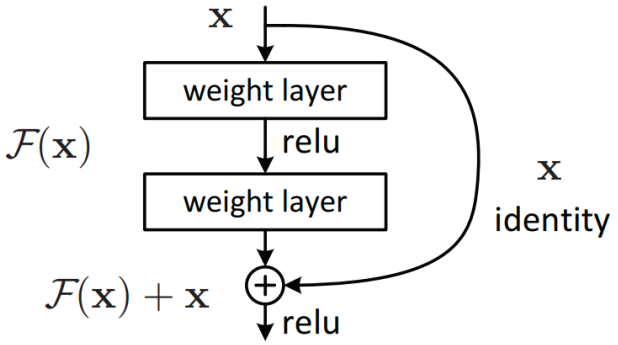

Figure 1. A block within a ResNet, taken from the original paper by He *et al*. [5]

word 'layer' is overloaded to denote both a set of blocks and the individual hidden layers within each block. The ResNet-101 model used in this project consists of a total of 33 blocks of 4 types of blocks, each with a different combination of channel size for the constituent convolutional layers. Due to computation efficiency considerations, ResNet-101, along with ResNet-50 and ResNet-152, differs from ResNet-18 and ResNet-34 in the use of Bottleneck blocks that reduce the number of channels of the input before expanding out again. This project used ResNet-101 to maintain consistency with the results reported in Xian *et al*. [].

4. Experiments

4.1. Dataset

The Caltech-UCSD Birds-200-2011 (CUB) dataset [14] contains 11788 images belonging to 200 bird species. The images were resized into 224x224 before fed into the ResNet model. Each image is annotated with attribute labels by multiple users of Mechanical Turk.

Attributes There are 28 attribute groupings and a total number of 312 binary attribute options, *e.g.* the attribute grouping *HasBellyColor* corresponding to 15 binary attributes each denoting one specific colour. Each attribute pertains to color, pattern, shape, or size of a particular body part. In my experiments, I ignored image-specific attribute labels and used class-aggregated attribute vectors. Each element in the aggregated attribute vector of a class is calculated as the percentage of the time that a Mechanical Turk user thinks the attribute is present for the class. Among the 28 attribute groupings, only 10 were normalised (binary attribute within the same grouping summing to unity). I normalised each grouping and used the same set of attribute encoding during training and testing.

Part	Attributes	Part	Attributes	Part	Attributes
Beak	*HasBillShape, HasBillColor , HasBillLength*	**Back**	*HasBackColor, HasBackPattern*	**Breast**	*HasBreastPattern, HasBreastColor*
Belly	*HasBellyPattern, HasBellyColor*	**Fore-head**	*HasForehead Color*	**Bird** (all parts)	*HasSize, HasShape*
Throat	*HasThroatColor*	**Nape**	*HasNapeColor*	**Head**	*HasHeadPattern*
Crown	*HasCrownColor*	**Eye**	*HasEyeColor*	**Leg**	*HasLegColor*
Tail	*HasUpperTailColor, HasUnderTailColor, HasTailPattern, HasTailShape*	**Wing**	*HasWingPattern, Has WingColor, HasWingShape*	**Body**	*HasUnderpartsColor, HasUpperPartsColor, HasPrimaryColor*

Figure 2. The total 28 attribute groupings, taken from Wah *et al*. [14]

Dataset Splits I used the splits suggested in Xian *et al*. [18] rather than the standard split, because there is an overlap of 43 images between CUB and ImageNet, the latter being the dataset used in the pre-training of ResNet. Although this is less than 1% of the size of test set, Xian *et al*. [18] observed that the models they tested consistently achieved higher accuracy on overlapping test classes.

4.2. Experimental Setup

Baseline For comparison, I used as input embeddings the 2048-dimensional output of pre-trained ResNet-101 both with and without finetuning on the CUB classes.

Training embedding extractors When training the attribute embedding extractors, the training targets were normalised continuous vectors instead of a single class label, hence the cross entropy loss was replaced by multi-label soft margin loss. I used the learning rate found for the ResNet-101 finetuned on CUB classes for all the attribute embedding extractors as well. The learning rate was set using discriminative fine-tuning [6], where the final layer was trained the learning rate found and each earlier layer had a lower learning rate, which decreases towards the input. The appropriate learning rate was found to be 10^{-3}. The embedding extractors were trained with the Adam optimiser for 10 epochs.

Training generalised ZSL classifier Early stopping was adopted in training the ZSL classifier, or more precisely, in training the mapping matrix W. A validation set was first split out of the training set to select for the appropriate

number of epochs and was merged back to the training set for formal training.

Evaluation I adopted the same evaluation protocol for generalised ZSL as Xian *et al*. [18]. I measured single label classification accuracy considering only the top-1 prediction. To account for class imbalance, macro-averaged accuracies for both seen and unseen test classes were reported, as well as their Harmonic mean. To measure the intrinsic quality of the attribute embeddings during training, I computed the relative frequency that the attribute probability vector predicted by the embedding extractor and the ground truth attribute vector share maximum element at the same index, *i.e.* the prediction is judged as accurate when the most likely attribute option predicted coincides with aggregated human judgements. I denote this intrinsic measure as top-1 max-probability accuracy and used it for choosing attribute embeddings.

4.3. Results

Choosing attribute embeddings I extracted attribute embeddings of two dimensions: 64 and 256. The top-1 max-probability accuracies obtained by the two versions of attribute embeddings, aggregated by attribute type, are shown below in Table 1.

attribute type	attribute count	test seen		test unseen	
		64-d	256-d	64-d	256-d
colour	16	69.2	71.2	62.1	62.5
pattern	6	77.8	79.1	71.9	72.1
shape	4	81.4	81.7	78.0	79.2
size	2	89.5	81.7	85.2	85.0

Table 1. Top-1 max-probability accuracies of 64-d and 256-d attribute embeddings. Attribute count denotes the number of attribute groupings within a type, *e.g.* there are 4 attribute groupings {bill-, tail-, wing-, overall-} of type *shape*, each consist of multiple attribute options. Individual accuracy for each attribute groupings can be found in the appendix

Although it seems that colour attributes are more difficult to predict than other types of features, it is noted that the options within each colour attribute grouping are also much more numerous (14 options for eyes and 15 for all other body parts).

As there is not much difference between the intrinsic evaluation measure of the 64-d and 256-d attribute embeddings, I used the former as input embeddings that is more computationally efficient for ZSL evaluation.

As an additional representation of the input image, I extracted the probability prediction for the attributes. In other words, each image is represented as a 312-dimensional vector of attribute probability. The accuracies achieved by these two different attribute-based representations as compared to the ResNet baselines are shown in Table 2.

Visualising the compatibility matrix When using attribute-based input embeddings, particularly attribute probability prediction vectors, the compatibility matrix introduced in section 3.1 can be used to interpret the behaviour of the ZSL model. Since attribute probability prediction vectors gave empirical results similar to those of attribute embeddings, I have focused on analysing the former here since the corresponding compatibility matrix is more interpretable. For clarity of visualisation, I avoided plotting the heatmap for the entire 312x312 matrix but only showed elements corresponding to non-colour attributes. The dimensions of the attribute probability prediction vector, $\theta(x)$, exactly match those of the output embedding, $\phi(y)$, and as shown in Figure 3, the diagonal elements were consistently of high positive magnitudes and sometimes surrounded by negative elements. This is not a surprise. Recall that the compatibility matrix maps the input embedding to the output embedding space. Given the projected input embedding $\hat{\phi}(y_{pred}) = W^T\theta(x)$, the diagonal elements of W correspond to the weights that connects the matching attributes in $\hat{\phi}(y_{pred})$ and $\phi(y)$, while elements close to the diagonal represent weights linked to different attribute options with the same attribute grouping. It is expected that the former correlates positively while the latter negatively. In fact, if ideal attribute probability prediction vectors can be extracted that perfectly match the averaged human judgments, W would be optimised to be an identity matrix.

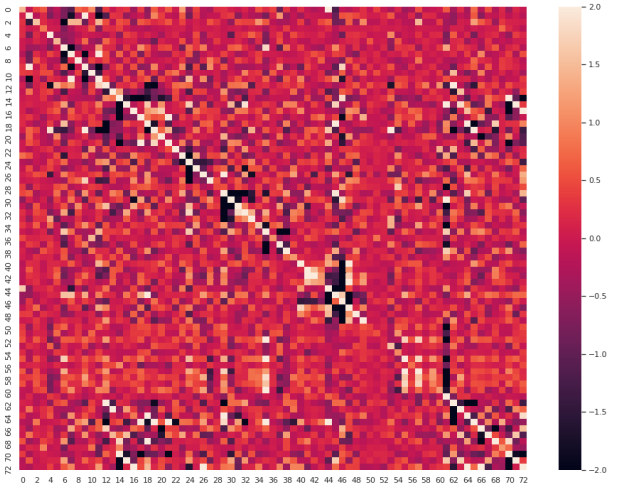

Figure 3. (Selected) compatibility matrix W for attribute probability prediction vector (corresponding to rows) as input embeddings and class attribute obtained by human judgement (correponding to columns) as output embeddings

Filtering cross-attribute correlations The aforementioned observation from the visualisation of W inspired me to explicitly remove the correlations between attribute groupings. During training, elements in W that describes the correlations between attributes within the same attribute grouping were maintained and all other elements were zeroed out, *i.e.* filtered, after each update of W. This yielded a much cleaner compatibility matrix (Figure 4) as well as better accuracies on unseen test classes (Table 2).

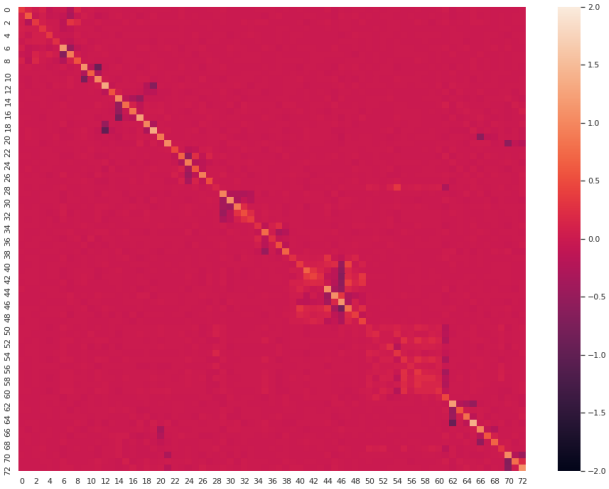

Figure 4. (Selected) compatibility matrix W with cross-attribute correlations filtered out

Quantitative evaluation of input embeddings As can be seen from Table 2, using either attribute embeddings or probability predictions can significantly boost the accuracy on unseen test classes, with the best harmonic mean obtained with attribute probability prediction vectors as the input embedding and cross-attribute correlation removed from the compatibility matrix. Although ResNet-101 achieved better accuracy on seen classes, the accuracy for unseen classes degraded from the ResNet-101 representations generated without fine-tuning. This suggest that while the representations were directly fine-tuned to optimise class prediction, the model can potentially overfit the training data, which gives the generated representations no advantage in the prediction of unseen classes.

Looking at the training curves of the ZSL classifier in Figure 5, we can see that the model that uses ResNet representation as input embedding converged to a much lower validation accuracy than when using attribute-based embeddings, despite the fact that ResNet representations had the largest dimension among the three options. Although unadapted SJE took much more epochs to converge when trained on attribute probability prediction vectors, the model converged much quicker using the same input embeddings when cross-attribute correlations were filtered out. The filtered compatibility matrix also enabled attribute probability prediction vectors to achieve similar performance as the more high-dimensional and hence computationally-expensive attribute embeddings.

input embedding	dim.	accuracy			
		training	test		
			seen	unseen	H
ResNet-101	2048	88.65	52.09	18.42	27.22
tuned ResNet-101	2048	93.08	**71.81**	17.59	28.26
attribute embedding	1792	95.21	66.99	30.65	42.05
attribute probability	312	94.41	65.39	28.60	39.79
attr. emb. (filtered)	1792	93.72	65.78	32.39	43.41
attr. prob. (filtered)	312	92.58	66.10	**32.69**	**43.75**

Table 2. The top-1 accuracies achieved with different input embeddings (H denotes the harmonic mean of accuracies for seen & unseen)

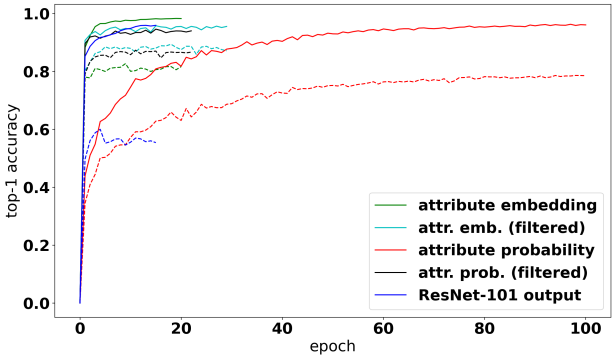

Figure 5. Top-1 accuracy throughout training epochs of ZSL classifier on training (solid lines) and validation (dashed lines) sets using different input embeddings

5. Discussions

In this study with the CUB fine-grained classification dataset, I found that attribute-based embeddings can achieve much better top-1 accuracy in predicting novel classes with only attribute information under the setting of generalised zero-shot learning, without significantly sacrificing the accuracy for seen classes. Admittedly, obtaining these attribute-based embeddings requires training multiple attribute-specific models, but these embeddings can have much lower dimensions which would speed up the training of the mapping-based zero-shot classifier, and more importantly, they could make the classification more efficient since the number of elements in the compatibility would be orders-of-magnitudes smaller. Acting as a preliminary study into this unexplored formulation of the input embeddings, this work calls for further investigation, in particular, to select for attributes that are most discriminative and to

identify and remove noisy features.

Acknowledgement The models used in this project were trained using GPU machines provided by the department.

References

[1] Zeynep Akata, Scott Reed, Daniel Walter, Honglak Lee, and Bernt Schiele. Evaluation of output embeddings for fine-grained image classification. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2927–2936, Boston, MA, USA, June 2015. IEEE.

[2] Vittorio Ferrari and Andrew Zisserman. Learning Visual Attributes. page 8.

[3] Kai Han, Jianyuan Guo, Chao Zhang, and Mingjian Zhu. Attribute-Aware Attention Model for Fine-grained Representation Learning. *arXiv:1901.00392 [cs]*, Dec. 2019.

[4] Ju He, Jieneng Chen, Shuai Liu, Adam Kortylewski, Cheng Yang, Yutong Bai, Changhu Wang, and Alan Yuille. TransFG: A Transformer Architecture for Fine-grained Recognition. *arXiv:2103.07976 [cs]*, Mar. 2021.

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, Las Vegas, NV, USA, June 2016. IEEE.

[6] Jeremy Howard and Sebastian Ruder. Universal language model fine-tuning for text classification. *arXiv preprint arXiv:1801.06146*, 2018.

[7] Dat Huynh and Ehsan Elhamifar. Compositional Zero-Shot Learning via Fine-Grained Dense Feature Composition. page 12.

[8] Kun Duan, D. Parikh, D. Crandall, and K. Grauman. Discovering localized attributes for fine-grained recognition. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3474–3481, Providence, RI, June 2012. IEEE.

[9] 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, Mar. 2014.

[10] Tsung-Yu Lin, Aruni RoyChowdhury, and Subhransu Maji. Bilinear cnn models for fine-grained visual recognition. In *Proceedings of the IEEE international conference on computer vision*, pages 1449–1457, 2015.

[11] Edgar Schönfeld, Sayna Ebrahimi, Samarth Sinha, Trevor Darrell, and Zeynep Akata. Generalized Zero- and Few-Shot Learning via Aligned Variational Autoencoders. *arXiv:1812.01784 [cs]*, Apr. 2019.

[12] Ming Sun, Yuchen Yuan, Feng Zhou, and Errui Ding. Multi-Attention Multi-Class Constraint for Fine-grained Image Recognition. *arXiv:1806.05372 [cs]*, June 2018.

[13] Ioannis Tsochantaridis, Thorsten Joachims, Thomas Hofmann, Yasemin Altun, and Yoram Singer. Large margin methods for structured and interdependent output variables. *Journal of machine learning research*, 6(9), 2005.

[14] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011.

[15] Josiah Wang, Katja Markert, and Mark Everingham. Learning Models for Object Recognition from Natural Language Descriptions. In *Procedings of the British Machine Vision Conference 2009*, pages 2.1–2.11, London, 2009. British Machine Vision Association.

[16] Wei Wang, Vincent W. Zheng, Han Yu, and Chunyan Miao. A Survey of Zero-Shot Learning: Settings, Methods, and Applications. *ACM Transactions on Intelligent Systems and Technology*, 10(2):1–37, Feb. 2019.

[17] Xiu-Shen Wei, Jianxin Wu, and Quan Cui. Deep Learning for Fine-Grained Image Analysis: A Survey. *arXiv:1907.03069 [cs]*, July 2019.

[18] Y. Xian, C. H. Lampert, B. Schiele, and Z. Akata. Zero-Shot Learning—A Comprehensive Evaluation of the Good, the Bad and the Ugly. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(9):2251–2265, Sept. 2019.

[19] Yongqin Xian, Saurabh Sharma, Bernt Schiele, and Zeynep Akata. F-VAEGAN-D2: A Feature Generating Framework for Any-Shot Learning. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10267–10276, Long Beach, CA, USA, June 2019. IEEE.

[20] Ning Zhang, Jeff Donahue, Ross Girshick, and Trevor Darrell. Part-based R-CNNs for Fine-grained Category Detection. *arXiv:1407.3867 [cs]*, July 2014.

Appendix A. Attribute Embedding Extractor

attribute class		num. of options	Max Prob. Accuracy Train	Valid	test seen	test unseen
colour	wing	15	67.2	61.2	61.4	53.4
	upperparts	15	68.3	63.4	62.9	50.3
	underparts	15	78.3	72.2	70.5	74.3
	back	15	71.7	65.7	64.4	54.7
	uppertail	15	72.0	65.6	63.8	57.3
	breast	15	77.4	67.6	68.9	69.1
	throat	15	82.5	73.5	72.8	63.8
	eye	14	97.6	96.7	96.7	97.2
	forehead	15	81.3	73.2	73.4	53.4
	undertail	15	70.0	63.8	64.2	56.3
	nape	15	75.8	67.2	66.0	55.4
	belly	15	79.7	72.1	71.4	73.3
	primary	15	71.0	65.7	66.9	59.7
	leg	15	78.3	64.9	62.7	62.4
	bill	15	74.1	65.6	65.8	60.0
	crown	15	84.9	76.8	75.2	53.1
pattern	head	11	74.8	62.6	63.0	53.5
	breast	4	94.9	83.1	84.7	84.5
	back	4	91.4	79.1	82.1	72.3
	tail	4	92.9	73.7	76.9	65.5
	belly	4	95.2	87.7	85.7	86.4
	wing	4	94.9	78.5	77.1	68.9
shape	overall	14	92.6	91.0	89.8	91.8
	bill	9	93.3	84.6	82.2	76.3
	tail	5	82.9	76.9	75.6	73.5
	wing	5	85.4	70.8	78.0	74.1
length	bill	3	96.7	91.1	90.7	83.6
size	overall	5	92.9	89.6	88.2	86.8

Table 3. Top-1 max-probability accuracy achieved with 64-d attribute embeddings for each individual attribute grouping

attribute class		num. of Options	normalised	max prob. Accuracy Train	Valid	test seen	test unseen
colour	wing	15		72.9	63.6	64.3	51.3
	upperparts	15		72.5	65.3	64.7	51.6
	underparts	15		82.8	71.3	73.2	73.6
	back	15		77.1	68.3	67.4	56.1
	uppertail	15		76.7	67.3	65.7	55.9
	breast	15		80.9	70.4	70.5	68.3
	throat	15		85.5	74.1	75.1	64.6
	eye	14		97.9	96.7	96.9	97.5
	forehead	15		82.4	73.4	74.0	54.9
	undertail	15		76.5	66.4	66.5	57.5
	nape	15		77.7	67.8	68.6	54.8
	belly	15		82.5	72.7	74.7	73.0
	primary	15		77.5	68.8	68.4	63.2
	leg	15		84.5	67.0	65.2	61.7
	bill	15		78.6	68.0	67.3	61.2
	crown	15		88.7	76.9	76.4	54.3
pattern	head	11		81.7	65.4	66.3	51.5
	breast	4	✓	94.6	83.8	85.7	85.3
	back	4	✓	92.8	79.0	82.0	73.2
	tail	4	✓	87.0	75.3	77.0	65.4
	belly	4	✓	97.0	87.0	85.7	87.3
	wing	4	✓	95.3	78.6	77.8	69.6
shape	bill	9	✓	92.9	84.4	83.0	77.2
	tail	5	✓	79.4	75.0	74.8	73.5
	wing	5		86.2	78.4	77.4	73.7
	overall	14	✓	94.8	92.9	91.7	92.4
length	bill	3		97.6	90.5	91.0	82.4
size	overall	5	✓	93.7	89.0	88.1	87.6

Table 4. Top-1 max-probability accuracy achieved with 256-d attribute individual embeddings for each attribute grouping (the *normalised* column shows whether the attribute grouping was normalised in the original class attribute encoding)