DD2412 Reproducibility Project: Semantic-Guided Multi-Attention Localization for Zero-Shot Learning

Romina Carolina Arriaza Barriga
rcab@kth.se

Lingxi Xiong
lingxi@kth.se

Styliani Katsarou
stykat@kth.se

Abstract

The aim of this project is to re-implement methods discussed in the paper "Learning where to look: Semantic-Guided Multi-Attention Localization for Zero-Shot Learning"[1] from scratch and to empirically analyze the results. The objective of this paper is classification of instances by semantic-guided Zero Shot Learning for seen and unseen data, enhanced by focusing on discriminative parts of the images using attention. A baseline to the original model is also introduced, which is limited to the original image's global features and does not include any attention on discriminative parts. In this reproducibility project, we implement the proposed models from scratch, without having any code provided by the writers as a reference point.

1 Introduction

The objective of Zero Shot Learning (ZSL) is to learn a concept without being trained on any examples, an approach that is in contrast with conventional object recognition methods which necessarily need a large number of sample images in order to collect as many instances as possible for every class. Nevertheless, this limits the conventional recognition process to classifying out of already seen classes, although in real life scenarios there exist object classes with a limited number of instances that the model is not trained on and thereby, not able of successfully recognizing them. In addition, in some cases a thoroughly complete labeling of images can get very expensive as it requires the presence of experts holding the appropriate domain knowledge. ZSL is filling these gaps by generalizing standard recognition models to identify novel object categories without having access to any labels. Currently, the dominating method in ZSL is learning concepts in language and transferring them to the visual domain, mimicking the human ability of recognising objects out of mere descriptions.

A high level description of the key idea behind this method is discovering how an unseen class is semantically related to a set of seen classes by associating visual feature vectors, extracted by convolutional neural networks, with linguistic context of images encoded in a high dimensional space of attributes. An attribute (*e.g.*, has blue eye color) is an intrinsic characteristic that an instance, or a class possesses that is usually defined by human experts. The learning process boils down to obtaining the mapping functions of projecting visual feature representations and semantic representations to a shared space. After mapping a given input image to the semantic space, which is essentially the output of the ZSL model, inference is done by measuring the distance of the output vector to all class vectors. The deep convolutional networks used to extract the visual features of the images can be pre-trained, and the aforementioned mapping functions are optimized either by ridge regression loss or ranking loss on compatibility scores of two mapped features.

Generally, the ZSL approach can be improved by identifying the significant regions of an input image. This has been achieved by employing a zooming network that discovers significant regions in order to learn the representations of the discriminative features [2], or by combining two semantic constraints to supervise attribute prediction and visual-semantic embedding[3]. Nevertheless, these techniques are restricted to extracting the global features of the input image without combining them with local feature representation learning. The paper under study[1] is the first attempt of combining zero shot learning with attention,in order to focus on the discriminative parts of an input image.

Specifically, the novelty of this paper[1] relies on enhancing the Zero Shot Learning technique by jointly learning local and global features of an image, under the guides of semantic representations. All semantic guided Zero Shot Learning studies prior to this one, have been solely relying on global features. Local feature extraction is conducted by a weakly supervised multi attention localization mechanism which detects the most discriminative regions of an image guided by semantic descriptions. The multi attention mechanism is supervised by two losses which encourage inter-class separability of features as well intra-class compactness. The learning of the discriminative features under semantic guidance is governed by another two losses that ensure inter-class distinction of features as well as intra-class divergence. The model is trained by minimising all four losses under the scope of an end-to-end learning.

2 Methods

This section includes a brief description of the framework of the model, followed by a detailed explanation of the parts it consists of. The four loss functions used to train the model are introduced and explained at the end of this section as well as the inference techniques employed.

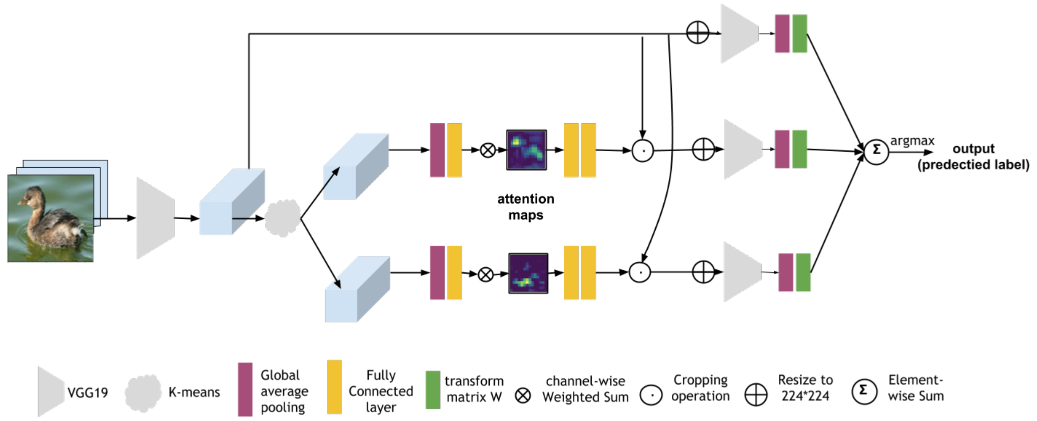

Figure 1: The Framework of the proposed Semantic-Guided Multi-Attention localization model (SGMA)

2.1 Multi-Attention Subnet

As depicted in Figure 1, the input image is initially fed into a pre-trained CNN following the VGG-19 architecture including up to the last average pooling layer. By exempting the last fully connected layers that form the classification part of the model, the pre-trained VGG-19 model acts as a feature extractor and provides with a feature map of size (H, W, C). K-means algorithm is then employed to cluster the channels of the feature map between two groups. Clustering relies on the logical hypothesis of grouping channels that establish similar or identical positions of peak values, thereby creating two groups of channels that correspond to two different parts of an image. These two groups, which are of same shape with the initial feature map derived by the CNN, will serve as initial predecessors of the final attention maps to be produced as an output of this first module. Figure 2 depicts the original image as well as the first and the second clusters derived after employing K-means, for visual inspection purposes. The two feature maps successfully highlight the two distinctive parts of this image, which are the tail and abdomen of the duck. The two generated clusters, go then through an average pooling layer that reduces the rank of the tensors by 2, by eliminating the (H, W) axes of each cluster, generating a vector p_c of length equal to the number of channels:

$$p_c = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} b_c(i, j) \tag{1}$$

where b_c is the feature in the c^{th} channel. The two generated p_c vectors are then fed into fully connected layers which produce the channel-wise attention weight vectors α_i:

$$a_i = \sigma\left(W_2 f\left(W_1 p\right)\right) \tag{2}$$

where $f(\cdot)$ refers to ReLU activation and i refers to the first and second attended part. The final step of this module is to compute the weighted sum between the channel-wise attention vectors and the initial feature vector as it was formed before K-means clustering:

$$M_i(x) = \sigma\left(\sum_{c=1}^{C} a_i^c f_{Conv}(x)^c\right) \tag{3}$$

$M_1(x)$ and $M_2(x)$ will be two attention maps of size (H, W) and c stands for their channels the total sum of which is C. The two attentions maps will then be used as an input for the second module of the model.

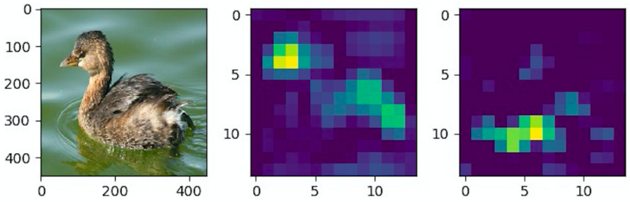

Figure 2: Left: Original Image; Middle and right: K-means derived clusters indicating the head and abdomen of the duck.

2.2 Cropping Subnet

This part receives the two attention maps as inputs and it is essentially "cropping" them in a differentiable way by a square with the highest value of each map as its center, following a similar approach as in [4]. In order to achieve this, it creates a two-dimensional continuous mask $V(x, y) = V_x \cdot V_y$ where

$$\begin{aligned} V_x &= f\left(x - t_x + 0.5t_s\right) - f\left(x - t_x - 0.5t_s\right) \\ V_y &= f\left(y - t_y + 0.5t_s\right) - f\left(y - t_y - 0.5t_s\right) \end{aligned} \tag{4}$$

where $f(x) = 1/(1 + \exp(-kx))$, t_s is the side length of the square, t_x and t_y denote the x-axis and y-axis square coordinates. Finally the cropped region is obtained by applying the mask to the initial image.

2.3 Joint Feature Learning Subnet

This is the final module of the network. It receives the original image as well as the part images as inputs. As shown in Figure 1, it passes the three images through three separate CNNs with the same VGG-19 architecture described in the Multi-attention part, to obtain their corresponding feature maps. It then produces the visual representation vectors of the original image and the two attended parts θ_0, θ_1 and θ_2 respectively, after feeding them to three separate average pooling layers first. Each visual feature $\theta(x)$ is mapped to the semantic space producing a compatibility score $s = \theta(x)^T W \phi(y)$ that is used as logit in the Embedding Softmax Loss. W is a trainable weight matrix which should have as many rows as the length of the visual feature vector $\theta(x)$, that is is equal to the number of channels of the feature map derived by the CNN, and as many columns as the number of attributes per image. Since ϕ has the shape of number of attributes times the number of classes, the resulting compatibility scores will be vectors of length equal to the number of classes. Since three different compatibility score vectors are produced by the model, they are summed up in order to compute the Softmax loss according to the late fusion strategy. This way there is no need of using a fully connected network after average pooling, thereby the total number of parameters is reduced.

2.4 Loss Functions

Compactness Loss is simply using the L_2 norm to support the attention map on focusing on one location instead of dispersing around the peak position:

$$L_{CPT} = \frac{1}{|\mathcal{Z}|} \sum_{z \in \mathcal{Z}} \|m_i^z - \widetilde{m}_i^z\|^2 \tag{5}$$

where z is the position of the attention map, $|\mathcal{Z}|$ is the size of the attention map, m_i^z and $\widetilde{m}_i^z$ denote the generated attention map and the ideal concentrated attention map at location z.

Diversity Loss makes sure that the generated attention maps highlight different location within the same image and avoid overlapping. As can be concluded by the mathematical expression of it, this function essentially boils down to an inner product between flattened matrices in order to measure the attention maps' similarity. By minimising this inner product, every location z that is of large value in one map is going to have a low value for the corresponding location of the other map and vice versa.

$$L_{DIV}\left(M_i\right) = \sum_{z \in \mathcal{Z}} m_i^z \max\left\{0, \widetilde{m}^z - mrg\right\} \tag{6}$$

where $\tilde{m}^z = \max_{k \neq i} m_k^z$ is the maximum of the other attention map at location z. The use of the margin is to ensure that this loss is less sensitive to noise.

Embedding Softmax Loss

$$L_{CLS} = -\frac{1}{N} \sum^{N} \log \frac{\exp(\boldsymbol{s})}{\sum_j \exp(\boldsymbol{s_j})}, j \in \mathcal{Y}_S \tag{7}$$

where $\boldsymbol{s_j} = \sum_i s_j^i = \sum_i \theta(x^i)^T W \phi(j)$, i stands for attention part. This loss function encourages inter-class distinction. The logits $\boldsymbol{s^j}$ are the compatibility scores derived in the third module of the model, and N is the number of training samples.

Class-Center Triplet Loss

$$L_{CCT} = \max\left\{0, mrg + \left\|\widehat{\phi}_i - \widehat{C}_i\right\|_2^2 - \left\|\widehat{\phi}_i - \widehat{C}_k\right\|_2^2\right\}, i \neq k \tag{8}$$

This loss is originally designed for intra-class visual feature distinction by minimising the distance between a baseline and a positive input and simultaneously maximising the distance between the baseline and a negative input. By principle, the implementation of this loss includes triplet sampling. Nevertheless, this loss is used in this paper in order to lower the intra-class divergence by substituting the baseline with the embedding $\widehat{\phi}_i$, the positive input with the correct class prototype $\widehat{C}_i$ and the rest of the class prototypes are considered as faulty inputs $\widehat{C}_k$. Consequently, there is no need for triplet sampling.

Total Loss The Joint Feature Subnet described in section 2.3 is trained under the objective of a total loss in a end-to-end fashion:

$$L_{SGMA} = \sum_{i}^{2} \left[L_{CPT}\left(M_i\right) + \lambda L_{DIV}\left(M_i\right)\right] + \alpha L_{CLS} + \beta L_{CCT} \tag{9}$$

2.5 Inference

There are two inference methods proposed. The first one is to straightforwardly choose the maximum value out of the compatibility score vectors obtained as outputs from the model, the index of which indicates the class. This is of more use for general zero-shot learning where the test set also contains the seen classes. The second inference method boils down to computing Φ_{cct}^u, the unseen classes-prototype matrix. Φ_{cct}^s is obtained by averaging the features of all images in every class. Taking into account that every image is described by 28 attributes,then it should have as many rows as the number of attributes and as many columns as the number of classes. Φ^u_{cct} can be derived by the assumption that the semantics of the descriptions of the unseen classes can be represented by a linear combination of those of the seen classes: $\Phi^u_{cct} = W\Phi^s_{cct}$ where W can be learned by solving the ridge regression as described in 10. Once Φ^u_{cct} is derived, then the classification is a matter of measuring the similarities between the two by computing the inner product: $\left\langle \phi_{cct}(x), [\Phi^u_{cct}]_y \right\rangle$.

$$W = \underset{W}{\arg\min} \|\Phi^u - W\Phi^s\|_2^2 + \lambda\|W\|_2^2 \tag{10}$$

Another inference method combines the two aforementioned methods by merely adding compatibility scores and the inner product between the semantics of the input image $\phi_{cct}(x)$ learnt, and the seen classes-prototype matrix Φ_{cct}: $y = \arg\min_{y\in\mathcal{Y}_\mathcal{U}} \left(s_y + \left\langle \phi_{cct}(x), [\Phi^u_{cct}]_y \right\rangle\right)$.

3 Experiments

3.1 Datasets

Two fine-grained medium-scale datasets, and one coarse-grained large-scale dataset commonly used for zero shot learning recognition tasks are used in this paper. **CUB-200-2011 dataset** [5] contains 11,788 images of 200 bird classes. All images are annotated with bounding boxes, part locations, and attribute labels. Images and annotations were filtered by multiple users of Amazon Mechanical Turk. Each sample is annotated with 312 attributes from 28 categories. **Oxford 102 Flower dataset** [6] is a collection of 102 common flower species each with 40 to 256 flower images, and 8,189 images in total. **Animals with Attribute (AwA)** [7] is a replacement dataset of the original *Animals with Attributes* dataset [8] which is not available anymore due to copyright restrictions. It consists of 37,322 images of 50 animals classes with pre-extracted feature representations for each image. A class/attribute matrix is included which provides with 85 numeric attribute values for each class. The CUB-200-2011 dataset is used in this report. Special train-test splitting method has to be used since the standard train-test split method violate the spirit of zero-shot learning that the testset has to be unseen data. Since the author didn't mention how they split the dataset, we follow the same settings in the paper they cited [7][9][10], where 27, 100, 62 training classes, 13, 50, 20 validation classes, and 10, 50, 20 test classes are used for AWA2, CUB, FLO, respectively.

3.2 Implementation Details

Our approach is implemented in Tensorflow 2.0. The implementation of our code has been made available [1]. If the parameter settings used by the authors were explicitly stated in the paper[1], we apply them to our approach. Otherwise, assumptions and experiments are carried out about the values of the ones that were not mentioned.

The VGG19 network used as a backbone, as mentioned in section 2, is a pre-trained one, but whether it is fine-tuned or not is not explicitly stated. The baseline paper[10] the authors are comparing with didn't fine-tune the feature extraction ResNet model. Consequently, we decided to use a pre-trained VGG19 model of freezed weights. Given the relatively small size of our dataset, not freezing the weights could lead to potential sabotage of the overall training process.The Multi-Attention Subnet input size is 448 by 448 and the same applies for the multi-attention-cropping subnet. The size of the input image for the joint feature learning Subnet has to be resized into 224 by 224 as stated by the authors. The chosen optimizer is SGD with learning rate of 0.05, learning rate decay by 0.1 on the plateau until $5 * 10^{-4}$, moment of 0.9, weight decay of $5 * 10^{-4}$. In Tensorflow 2.0, the SGD with weight decay can be achieved either by applying L2 regularizers on layers or through the SGDW function. The latter approach was chosen for our approach. The batch size is 32 trained on two GPUs(TitanX). For our approach, two GPUs(Nvidia V100) provided by Google Cloud were used. The margins used in equations 6 and 8 are set to 0.2 and 0.8 for L_{DIV} and L_{CCT}, respectively. In equation 4, k is equal to 10. As mentioned by the authors, in equation 3 i can either be equal to 1 or 2, given that more than 2 attention maps would result to overlapping of attended parts.

The next two sections focus on implementation details specifically for the baseline model and the final model.

[1] `https://github.com/LindsayXX/DD2412_project`

3.3 Baseline Model

Apart from the complete model described in the Methods-Section, that consists of all three modules, a basemodel was also implemented that only includes feature extraction by a pre-trained CNN of the VGG-19 architecture, average pooling and derivation of the compatibility score by employing the method described in section 2.3. Following the definition in the ablation study, only Embedding Softmax Loss is used to train the basemodel.

Initialization of W The transform matrix W is an important trainable variable that will be used for the inference model but its initialization is not specified. We tried different initialization methods: random normal initialization and uniform initialization. In the end, we follow one of the main reference[2] of the orignal paper that learning W through a fully connected layer, which is essentially the a linear project matrix that maps $\theta(x)$ to the semantic attribute space Φ. We set the activation to None and the default He initialization.

Semantic descriptions $\mathbf{\Phi}$ is a matrix that contains semantic feature vectors $\phi(y)$ of all classes that is used for visual-semantic embedding. The semantics of image refer to the attributes of the animals and flowers. According to the author, they followed the same attribute as provided in[7][10][9], where ten single-sentence visual descriptions for each image is encoded and a single averaged attribute vector per class was computed. In the end, we form a binary attribute vector for each sample and take the average of each class for the final $\mathbf{\Phi^s}$.

3.4 Final Model

Initialization of weights for the fully connected layers regarding the Multi-Attention Subnet as explained in the original paper, is of the form:

$$\hat{a}_i = [\mathbb{1}(1), \mathbb{1}(2), \ldots, \mathbb{1}(c), \ldots, \mathbb{1}(C)] \tag{11}$$

The initialization of the channel-wise attention weights for the two attended parts, where i indicates the part, boils down to an indicator function over feature channels where $\mathbb{1}(c)$ is equal to 1 to indicate that channel c belongs to ith part, or 0 otherwise. There is no indication as to how many epochs were used for the training process. Our model was trained for 30 epochs.

Initialization of the centers in the class-center triplet loss is not mentioned. We set it to Gaussian distribution of $\mathcal{N}(0, 0.01)$, as this is the used in the most cited class-center triplet loss paper [11].

3.5 Results

For the baseline model, with the same batch size of 32 and freeze the weight of VGG, we discover that the model is very hard to train. Lower learning rate(1e-5, 1e-6), gradient clip and normalization have been tried but it's still difficult to train. We think it is because of two many neurons in the fully connected layer with only one layer for training.

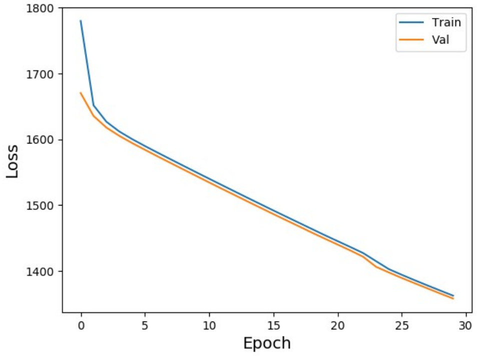

Figure 3: Loss reduction after 30 epochs for training and validation set

We prove the functionality of our implementation of the final model by loss decrease as shown in Figure 3. Kmeans clustering, as part of the Multi Attention Subnet, successfully produces two groups of channels that focus on different parts of the image, as can be seen in Figure 5.We can prove the efficient performance of the Multi Attention Subnet governed by L_{DIV} and L_{CPT}, by observing how the attention maps tend to get apart in order to obtain two local attentions after training. As depicted in Figure 4 the derived attention maps successfully focus on the discriminative parts of the image: wings(middle image) and head (on the right). Nevertheless more training would allow for the L_{DIV} to further separate the discriminative parts.

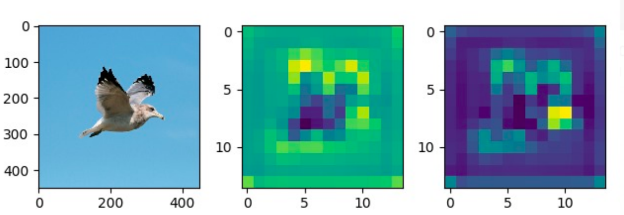

Figure 4: Left: Original Image; Middle and right: The derived attention Maps out of the Multi Attention Subnet after training

4 Conclusions and Discussion

4.1 Limitation of the paper and proposed methods

As part of giving constructive criticism, one could mention that the overall explanation of the methods used to construct the proposed model architecture is not coherent enough, since many ambiguous descriptions were detected, that could be confusing or even misleading. A representative example of such a case, is the provided explanation of the Joint feature learning Subnet: at the final part of the model that integrates global and local features, the late fusion strategy for the compatibility scores in embedding Softmax loss was employed by the authors. Nevertheless, the way of dealing with the mapped features was not mentioned, which lead to two alternative ways on how to define the transform matrix that follows. The definition of the class-center triplet loss was not detailed enough. Triplet-loss was originally designed for face recognition tasks but has also been used for learning similarity in learning embeddings. Nevertheless, when introducing class center triplet loss, authors neither cited any paper nor motivated enough to make it sound like an original idea.Also no adequate explanation of the centroids was given. The definition of triplet-center loss as mentioned in two related papers, [11] and [12], mentions the use of the minimum class difference instead of all classes for the negative center, which can be considered more reasonable as well as more feasible implementation-wise.

An interesting conclusion is that in some cases, the Kmeans clustering can indicate misleading details of pictures. Figure 5 shows such a case: one of the clusters focuses on the flower, which is not a discriminative feature of the bird. This implies that the K-means clustering may not be the ultimate technique to chose, as it could eventually limit the attention mechanism, and potentially lead it to failure cases.

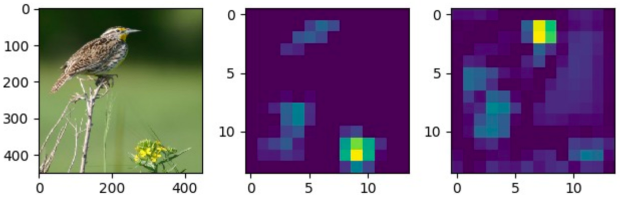

Figure 5: Left: Original Image; Middle and right: K-means derived clusters, misleadingly indicate the focus on the bird and an irrelevant object instead of two discriminative parts of the bird

4.2 Challenge of reproducibility

Reproducing the paper out of solely reading it can be considered as a task of high difficulty. Possibly due to the page limit, not all of the implementation details are mentioned in the original paper. Given that the code is not publicly available, we had to make speculations on key parts of the implementation, such as the initialization of fully connected layers,the transform matrix, the semantic centroids or how to split the train and test data, number of epochs for training and more details that are discussed in section 3. The construction of a model as big as the one proposed in the paper, requires a large amount of parameters. Not being provided with a considerable amount of these parameters' values, can convert a reproducibility process to a tedious fine-tuning challenge and thereby highly compromise the implementation process.

References

[1] Yizhe Zhu, Jianwen Xie, Zhiqiang Tang, Xi Peng, and Ahmed Elgammal. Learning where to look: Semantic-guided multi-attention localization for zero-shot learning. *arXiv preprint arXiv:1903.00502*, 2019.

[2] Yan Li, Junge Zhang, Jianguo Zhang, and Kaiqi Huang. Discriminative learning of latent features for zero-shot recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7463–7471, 2018.

[3] Pedro Morgado and Nuno Vasconcelos. Semantically consistent regularization for zero-shot recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6060–6069, 2017.

[4] Jianlong Fu, Heliang Zheng, and Tao Mei. Look closer to see better: Recurrent attention convolutional neural network for fine-grained image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4438–4446, 2017.

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

[6] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing*, pages 722–729. IEEE, 2008.

[7] Yongqin Xian, Christoph H Lampert, Bernt Schiele, and Zeynep Akata. Zero-shot learning-a comprehensive evaluation of the good, the bad and the ugly. *IEEE transactions on pattern analysis and machine intelligence*, 2018.

[8] Christoph H Lampert, Hannes Nickisch, and Stefan Harmeling. Attribute-based classification for zero-shot visual object categorization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(3):453–465, 2013.

[9] Scott Reed, Zeynep Akata, Honglak Lee, and Bernt Schiele. Learning deep representations of fine-grained visual descriptions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 49–58, 2016.

[10] Yongqin Xian, Tobias Lorenz, Bernt Schiele, and Zeynep Akata. Feature generating networks for zero-shot learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5542–5551, 2018.

[11] Xinwei He, Yang Zhou, Zhichao Zhou, Song Bai, and Xiang Bai. Triplet-center loss for multi-view 3d object retrieval. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1945–1954, 2018.

[12] Zhaoqun Li, Cheng Xu, and Biao Leng. Angular triplet-center loss for multi-view 3d shape retrieval. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 8682–8689, 2019.