Generalized Zero-shot Learning with Attention Mechanism

Muqiao Yang (muqiaoy) [1] **Jiaxu Zou (jiaxuz)** [1] **Chentao Ye (chentaoy)** [1] **Qingtao Hu (qingtaoh)** [1]

1. Introduction

Supervised classification algorithms have been very successful in applications of many fields, with its impressive performance and easy-to-train characteristic. In recent years, the rise of deep learning techniques boosted the performance even further. One of the largest limitations of typical supervised learning, nonetheless, is that it is only able to classify categories the model has seen before. Worse still, in order to learn patterns of these specific categories, the neural network model has to see sufficient amount of data from the categories.

Zero-shot learning (ZSL) (Lampert et al., 2009) has been an effective learning paradigm applied in the situation, where some test classes unseen previously during training are encountered during evaluation. The concept of ZSL is significant, because in the real world, we usually cannot collect sufficient data for each class, especially when the number of classes is huge. In other situations, we may even not be able to gain any relevant data for certain classes, because of expensive cost and difficult accessibility. The general idea of ZSL is to extract latent relationship shared among both seen and unseen classes, so that the classifier could assign the correct label to a newly seen object using this transferred knowledge. The concept comes from the intuition that humans not only identify objects of seen classes by their observable features, but also using existing knowledge.

However, most previous work (Frome et al., 2013) about ZSL is evaluated only on samples of unseen classes. (Xian et al., 2018a) proposed a generalized zero-shot learning (GZSL) setting, where classes for test can be either seen or unseen, which is more similar to real world applications. It is validated that results of most ZSL algorithms under GZSL setting become significantly lower than evaluated under traditional ZSL setting, because seen classes in the search space could distract the model from recognizing images correctly.

In this paper, we focus on developing a novel model in the problem settings of GZSL, which achieves better results compared to previous methods.

2. Literature Survey

Researchers in ZSL community mainly focus on three tasks:

1. Learn the mapping from image space to feature space.
2. Learn the representation of class in the semantic space.
3. Learn the mapping from feature space to semantic space.

The first task is closely related to deep-learning based feature engineering, which is a classical task in computer vision community. A large number of achievements have been made using different types of convolutional neural network (e.g., ResNet (He et al., 2016), VGG16 (Simonyan & Zisserman, 2014), DenseNet (Huang et al., 2017)). There hasn't been a mature pipeline, nevertheless, for the second and third task. These two tasks have gained significant attention from research communities in recent years. Although these two tasks are equally important for ZSL, the main division of existing ZSL methods depends on how the second task is addressed, i.e., the representation of class in the semantic space. There are three common representations: **attribute description vectors**, **class embeddings** and **entities in the knowledge graph**.

The first paper on ZSL defined and formulated the task, as well as proposed one solution of class representation using attribute descriptions (Lampert et al., 2009). Each class has an attribute description vector, which is a list of identifications for several salient attributes of that class (e.g., "black skin", "eats grass", "has stripes" for zebras). Each identification clarifies whether the class has such an attribute. The paper proposed two methods of modeling the mapping between the feature space and the semantic space: Direct Attribute Prediction (DAP) and Indirect Attribute Prediction (IAP). Although the author does not leverage deep learning model architecture to form the mapping, the concept of knowledge transfer from low-level visual features to semantic features is well illustrated.

The benchmark model of building semantic space based on class embeddings was developed in 2013 (Frome et al.,

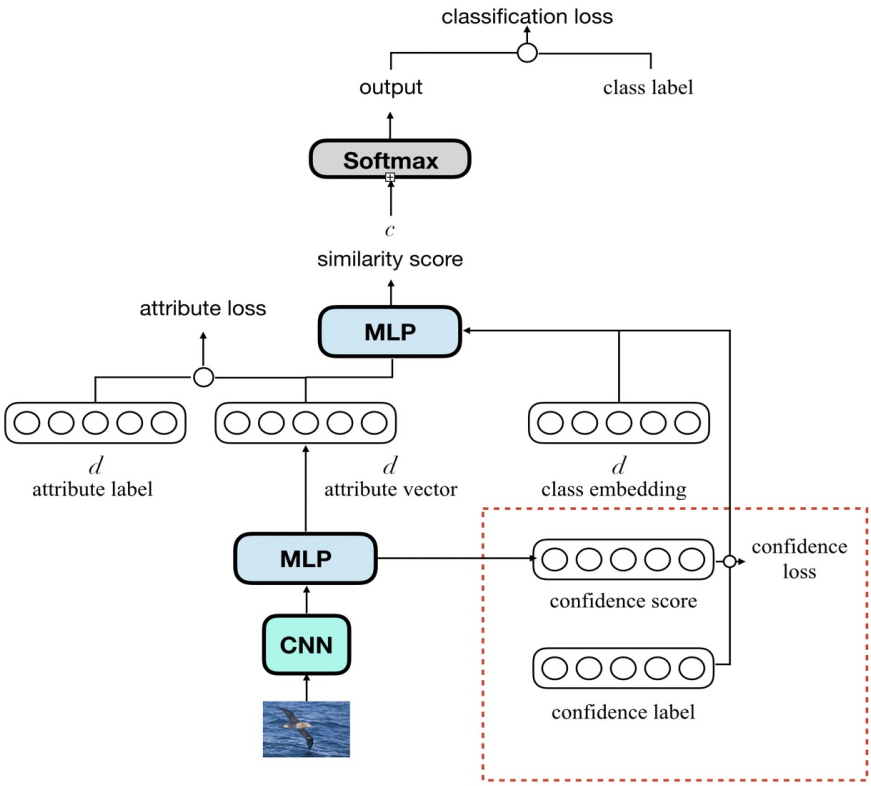

Figure 1. Framework of Base Model

2013). From the concept, each class is represented by a word vector, which could be trained using Skip-Gram or CBOW model like pre-trained word embeddings in NLP tasks. Then the visual feature vectors are projected to the semantic space with the same dimension as pre-trained class embeddings via a transformation layer. Then the model computes the similarity between projected vectors and class embeddings. Finally, the label of the class with the highest semantic similarity with the visual features is assigned to the image.

While there are many approaches to do zero-shot learning with only image inputs, there will be more helpful if we also have semantic information about classes (Xian et al., 2018b). For example, we can train a GAN which synthesizes CNN features conditioned on the class descriptors to avoid cascading errors along the original process. To achieve that, a Wasserstein GAN and a classification loss are enough to learn the essential feature distributions about each class given its semantic information.

Another intuition of solutions to the problem is to determine the similarity, or relevance between query images, to help determine the class of asked instance. A relation network can be trained to learn a deep distance metric (Sung et al., 2018a). In that process, only very few instances need to be provided, and we can input query images and images of rare classes to get the relation distance. That will not only help determine the rare classes, but also give the model a general feeling about closeness of instances. Although this is not strict zero-shot learning, this approach is very intuitive and it is not hard to extend its power in the scenario of zero-shot learning.

Attention mechanism is widely applied in sequence-based models (Bahdanau et al., 2014), aiming to focus the attention of the model on the most important part of the sequence. Most of existing ZSL algorithms use all visual features in the image to train the model, where much irrelevant information for classification is incorporated and may dampen the performance of the model. We thus propose to use part locations to split the image into local feature regions. Since we have the semantic representations of classes, it is useful to learn the mapping from feature space to semantic space with attention to most relevant features for zero-shot classification. An attention mapping is learned to map the relevance of local regions to representations in semantic space.

Fully aware of the fact that ZSL is still a relatively new topic without benchmark of evaluation metrics, we need to use the same evaluation metrics as used in previous work which we want to compete with. This way, it is much easier to compare the performance of different models and further analyze the benefits and flaws of them.

3. Proposed Method

3.1. Base Model

(Sung et al., 2018b) uses an embedding module to map both input images and class embeddings to a latent space with the same dimension, and then uses a relation module to concatenate two latent vectors and compute the final relation score. In this approach, the intermediate representations in the latent space lose specific meanings. We want to explicitly represent the semantic information in the latent space, because only in that way does the relation matching done in the second module make sense in terms of explainability. We take explicit attribute descriptions for each image as our intermediate representations. We keep the relation module to compute a relation score for each **class embedding - attribute description** pair.

The architecture of our base model is shown in Figure 1 (the part inside the red box is related to calibration with confidence, which will be stated in Section 3.2). Our model basically consists of two modules: **Attribute Module** and **Relation Module**. Attribute Module maps the original image to attribute description representation. It contains a Convolutional Neural Network (CNN) as a sub-module to extract low-level features from the image. Commonly used CNNs are ResNet (He et al., 2016), DenseNet (Huang et al., 2017), etc.. The low-level features are fed to a Multilayer Perceptron (MLP) to be converted to an attribute description vector. In Relation Module, the attribute description vector and the class embedding vector are concatenated and fed to another MLP to obtain a relation score. For training, we use the attribute loss for Attribute Module and the classification loss for Relation Module, separately. For inference or test, the attribute description generated by the Attribute Module is fed to Relation Module in pair with every class embedding in the test set. The class with the highest relation score is the predicted class.

3.2. Calibration with Confidence

At inference time, some attributes may not be visible in certain images. (E.g., the tail of a bird is blocked by a brick as shown in Figure 2). In such case, the attribute value does not make any sense and we do not want to take it into consideration. To address the case, we incorporate the mechanism of calibration with confidence to our base model. The confidence is defined as how much the model believes in its predicted attribute values. We make Attribute Module generate an additional confidence vector with the same dimension as the attribute description vector. Then what Attribute Module does is to predict an attribute description vector for each image and indicate to what extent each attribute value is useful for classification of the image.

Figure 2. Sample images where some important features (like tail) are occluded.

In Relation Module, we propose several approaches to applying calibration with confidence to the predicted attribute description vectors.

3.2.1. Hard Masking

We manually set a threshold for confidence scores. If a confidence score is lower than the threshold, the corresponding attribute value should be discarded. But we cannot remove those elements from the vector because the input for Relation Module is required to be a fixed size. We assume that the original attribute description vector has d elements and we will discard n low-confidence elements. We want to mask those values by blocking n neurons in the first layer of Relation Module. In order to block partial neurons without destroying existing semantic meanings of other attribute values, we borrow the idea of Dropout in deep neural network (Srivastava et al., 2014). The idea is to scale up all weights in the first layer by a factor $1/\alpha$ after setting the value of n neurons to 0, where

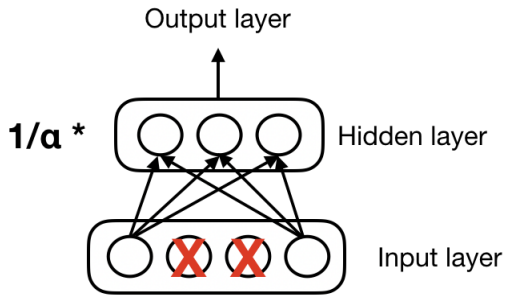

Figure 3. A toy example of mechanism masking out trivial features (denoted by red cross). The value of the first hidden layer would be scaled by $\frac{1}{\alpha}$ to balance the network, where α is the fraction of features that are masked out. The idea is borrowed from dropout (Srivastava et al., 2014).

$\alpha = \frac{d-n}{d}$, shown in Figure 3. Intuitively, the operation masks non-confident attribute values by enhancing other more confident attribute values. In the previous example, it means that our model makes the decision relying more on the visible attributes (special wings and feathers) but not the occluded attributes (tails).

3.2.2. Soft Masking

Instead of manually setting a fixed threshold for confidence scores, a "softer" way is to pass the predicted attribute description vectors and confidence vectors to Relation Module. The idea is to make the model learn such "threshold" or "masking criteria" from its own classification performance during training. In other words, we integrate the masking process into an End-to-End network to encourage fine-grained and adaptive processing, which may improve the model to some extent.

We modify the model architecture accordingly (see the red box in Figure 1). For both hard masking and soft masking, Attribute Module outputs an additional confidence vector and is trained with an additional loss, which is confidence loss. Relation Module takes in an addtional confidence vector concatenated with the attribute description vector and the class embedding.

3.3. Multi-Attention Model

Attention mechanism has been applied to a wide range of computer vision applications, including image classification (Xiao et al., 2015) and semantic segmentation (Chen et al., 2016). Generally, the concept of an attention model is to adaptively locate the most relevant information to the task in the input (Wang et al., 2017). Here, we

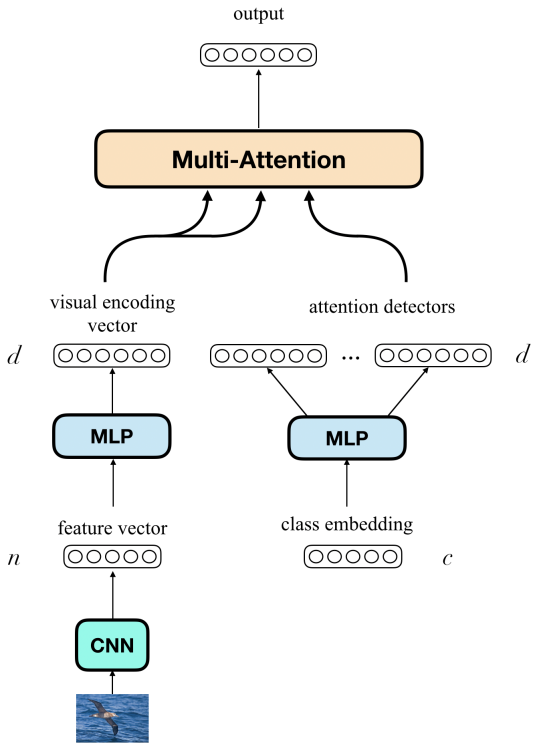

Figure 4. Model structure for Multi-Attention Model

believe that ZSL task will definitely benefit from the attention mechanism, since human recognition of images also involves such mechanism, where some features in the image provides much more information for classification than others. When more and more classes are considered, capturing the difference of such key features between classes undoubtedly helps classification. We apply the mechanism here by making the model generate attention detectors from the class embedding, and making it learn an attention map from those detectors used for classification.

The model architecture is shown in Figure 4. Same with the base model, we also pass the raw images through a CNN to extract low-level features. Then we apply a local feature encoder to get the visual feature encoding vectors. Then the weighted pooling of these visual encoding vectors $\{\mathbf{v}_i\}_{i=1}^n \in \mathbb{R}^d$ is calculated under a set of attention maps $\{a_i\}_{i=1}^n$ to generate attention vector $\mathbf{k} \in \mathbb{R}^d$, where n is the number of visual encoding regions in one image. The operation is defined as

$$\mathbf{k} = \sum_i a_i \mathbf{v}_i, \ \mathbf{k} \in \mathbb{R}^d$$

For the attention weights a_i, firstly the attention detectors

$\mathbf{h} \in \mathbb{R}^d$ are generated from the class embedding $\mathbf{c}$:

$$\mathbf{h} = \mathbf{W}\mathbf{c} + \mathbf{b}$$

where $\mathbf{W} \in \mathbb{R}^{d \times c}$ is the weight matrix and $\mathbf{b} \in \mathbb{R}^d$ is the bias, where c is the dimension of the class embedding. Then the attention weights are calculated as

$$a_i = \frac{\text{ReLU}(\mathbf{h}^T \mathbf{v}_i)}{\sum_i \text{ReLU}(\mathbf{h}^T \mathbf{v}_i)}$$

The attention values are normalized. Here ReLU function is applied after the dot product of $\mathbf{h}$ and $\mathbf{v_i}$, with an intuition that the irrelevant information in the original image whose attention weights are negative should be ignored.

Furthermore, we also introduce the concept of multi-attention by generating multiple sets of attention maps at one time (Vaswani et al., 2017). Typically, the objective of multi-attention is to describe an image from several perspectives with different parts and contexts highlighted. Additionally, it can reduce the risk of attending to incorrect regions compared to using a single attention map.

By using multi-attention mechanism, we generate attention vectors $\{\mathbf{k_j}\}_{j=1}^M$ where M is the number of attention heads. The final aggregated attention vector K is defined as:

$$K = [\mathbf{k}_1 \ \mathbf{k}_2 \ \cdots \ \mathbf{k}_M]$$

4. Experiments

4.1. Dataset

To evaluate our model, we use Caltech-UCSD-Birds 200-2011 (CUB 200-2011) (Wah et al., 2011) dataset. The dataset contains 11,788 images of birds from 200 species. 150 species are used as seen classes during training and the remaining 50 are used as unseen classes during test. Under GZSL setting, test images should cover both seen and unseen classes, so we combine the images of the 50 unseen species with part of the images of 150 seen species. Thus we have 7,057 training images covering 150 seen species and 4,731 test images including 2,967 images from 50 unseen species and 1,764 images from 150 seen species.

Along with the images and class labels, the dataset also provides a 312-dimension class embedding for each of 200 classes. The class embedding is trained like word embeddings on a large corpus in which the texts are related to the corresponding class (Reed et al., 2016). It implicitly encodes semantic information about the class. Besides, the dataset provides a 312-dimension attribute description vector for each image. Each element of the vector represents an attribute of the bird (e.g., the colour of the feather, the shape of the beak, etc.) and they only take binary values. "0" means such attribute does not exist while "1" means the opposite. It is noteworthy that attributes describing the same part of the bird are always exclusive. Additionally, a certainty for each attribute value is provided from statistical results of certainties given by Amazon Turkers when they annotated the data. The certainty takes 4 discrete values, indicating 4 levels of confidence.

4.2. Metric

We have evaluation metrics for both traditional ZSL and GZSL. We first define per-class accuracy Acc_C as the following:

$$Acc_C = \frac{1}{|C|} \sum_{i=1}^{|C|} Acc_{C_i}$$

$$Acc_{C_i} = \frac{\#(pred(x_{C_i}^{(j)}) == y_{C_i}^{(j)})}{|C_i|}$$

where Acc_{C_i} is the classification accuracy of the model on test samples of class C_i. Per-class accuracy is the average of accuracy on all classes.

For traditional ZSL metric, we evaluate the model only on unseen classes of test samples. We calculate the per-class accuracy of the model on a U-way classification task, where U is the number of unseen classes. The metric is denoted as **ZSL-T1**.

For GZSL metric, we evaluate the model on both unseen classes and seen classes of test samples. We first calculate the per-class accuracy on unseen classes for a C-way classification task **GZSL-U**, where C is the total number of the unseen and seen classes in test data. We then calculate the per-class accuracy on seen classes for a C-way classification task **GZSL-S**. We finally compute the harmonic mean of $GZSL-U$ and $GZSL-S$ to get the final score **GZSL-H**.

$$GZSL-H = \frac{2 \times GZSL-U \times GZSL-S}{GZSL-U + GZSL-S}$$

4.3. Hyperparameters

For the base model, we use a 2-layer MLP for both Attribute Module and Relation Module. For Attribute Module, the size of the hidden layer is 1200 and the size

Models	ZSL-T1	GZSL		
		U	S	H
CONSE	34.3	1.6	**72.2**	3.1
DEVISE	52.0	23.8	53.0	32.8
SYNC	**55.6**	11.5	70.9	19.8
ESZSL	53.9	12.6	63.8	21.0
SAE	33.3	7.8	57.9	13.7
DEM	51.7	19.6	54.0	28.8
ALE	54.9	23.7	62.8	**34.4**
BM	47.9	17.3	49.8	25.7
BM (SRM)	48.6	17.7	49.2	26.0
BM (SRM)(End-to-End)	52.7	22.3	59.9	32.5
BM (SRM) + CC (Hard)	47.6	17.5	51.9	26.2
BM (SRM) + CC (Soft)	48.8	17.8	53.3	26.7
BM (SRM) + CC (End-to-End)	53.5	22.7	60.2	33.0
Multi-Attention Model	44.3	**24.5**	51.9	33.3

Table 1. Performance of Models under different ZSL metrics

of the output layer is 312. For Relation Module, the size of the hidden layer is 300 and the size of the output layer is 1. We choose the same CNN as in (Sung et al., 2018b), which is ResNet-101 pretrained on ILSVRC 2012 1K classification without fine-tuning. We take the top pooling units as low-level feature vectors with dimension of 2048 for each image. For activation function of all the hidden layers, we use ReLU. We apply Sigmoid activation to the output of Attribute Module and construct binary cross-entropy as the attribute loss. We apply Softmax to the output of Relation Module and construct T-way cross-entropy as the classification loss, where T is the number of classes in the training set (i.e., the number of seen classes). We separately train Attribute Module and Relation Module. We use Adam optimizer for both modules. We start with an initial learning rate of 1e-4 and decay it by 0.5 for every 30000 iterations. We train the network for 1000 epochs and saved the model with the best test accuracy.

For hard masking of calibration with confidence, we use the threshold of 0.5 to filter confidence values. Other hyperparameters are kept the same with the base model.

For Multi-Attention Model, we set M, the number of attention heads to 4. Other hyperparameters are kept the same with the base model.

4.4. Results

All of our results, compared to several previous representative methods (Norouzi et al., 2013), (Frome et al., 2013), (Changpinyo et al., 2016), (Romera-Paredes & Torr, 2015), (Kodirov et al., 2017), (Zhang et al., 2017) and (Akata et al., 2016) are shown in Table 1. (In the table, **BM** is short for **Base Model**; **SRM** is short for **Stochastic Relation Matching**; **CC** is short for **Calibration with Confidence**.) The percent sign is omitted in the table.

4.4.1. Base Model

From the results, our base model has a fairly good performance, beating all the other previous methods except **DEVISE**, **DEM** and **ALE**. Compared to these 3 methods, our base model is defeated under all 4 metrics, so it seems that there is large room for improvement of the base model.

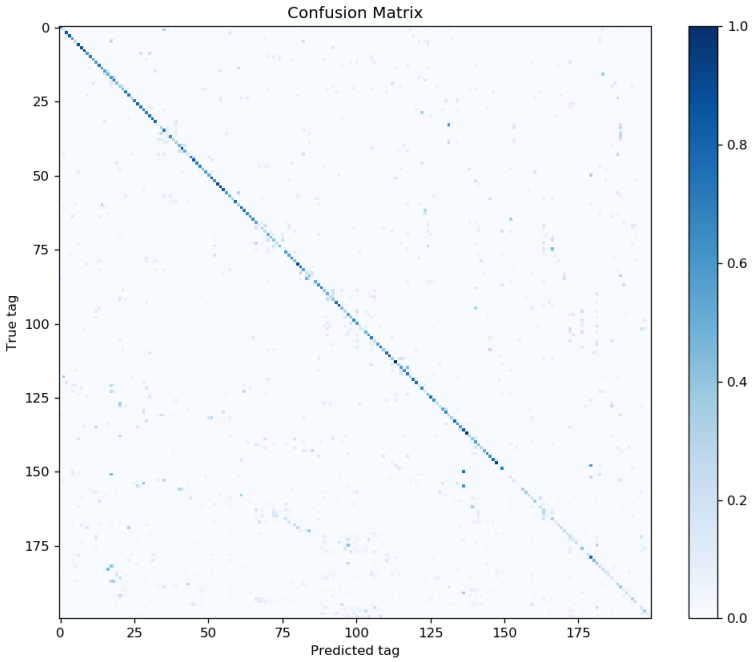

Figure 5. Confusion Matrix of Multi-Attention Model

4.4.2. Stochastic Relation Matching

When we look at the training process, we find that at early stages, the accuracy increases quite slowly compared to later stages. The reason may be that 150-way classification is too difficult for the model to learn to perform. Inspired from Stochastic Gradient Descent (SGD) (Robbins & Monro, 1951), we propose performing stochastic relation matching (SRM) at training time. Instead of using 150-way cross-entropy as our classification loss for Relation Module, we only consider the classes which exist in each batch and construct a C-way cross-entropy as the classification loss, where C is the number of classes involved in each batch. Since we have $C \leq$ batch size, C-way classification is much easier for the model to perform. The advantage is that the model will capture more fine-grained similarity/difference between classes as if it used a magnifier to observe local objects. Since the involved classes vary from batch to batch, the shifting average of this stochastic version is a good estimate of the original one. The results show that SRM really improves the base model by a little bit. GZSL-H increases from 25.7% to 26.0%. We keep using SRM in our later experiments.

4.4.3. End-to-End Training

End-to-End training is a popular technique in deep learning community. Basic idea is that our prior or inductive bias about the model architecture will sometimes keep us from achieving the optimal performance. Therefore, instead of keeping several modules with different functions, we treat the whole neural network as a blackbox and make the model learn its best weights to fit the task by searching the huge function space. Here we remove the attribute loss and train both modules at the same time by backpropagating the classification loss through the whole network, regardless of the semantic meaning of the vector in the intermediate space. It is surprising that both GZSL-U and GZSL-S have a remarkable increase. It shows that having explicit attribute description representations in the intermediate space does not help the model that much.

4.4.4. Calibration with Confidence

From the results, hard masking of calibration with confidence improves the model by 0.2% and soft masking improves the model by 0.7%. Actually, soft masking is like an End-to-End version of hard masking, which is expected to work better.

We also propose a more general End-to-End version of calibration with confidence. We do not even need to explicitly represent confidence vectors. We just concatenate the class embedding to the two vectors generated by Attribute Module and pass the three vectors to Relation Module. We see another significant improvement in terms of all metrics. The result again show the strengths of End-to-End training for improving the model.

Generally there is no much improvement brought by calibration with confidence. The main reason is that the confidence information is provided by Turkers, which will have big variance if there are very few Turkers. Another reason is that in most cases the attribute value has high confidence and calibration is not necessary at all.

4.4.5. Multi-Attention Model

We achieves our best result using Multi-Attention Model, which is highlighted in Table 1. Although on seen classes, there is little improvement compared to other models, Multi-Attention Model outperforms others on most unseen classes, and achieves a higher GZSL-H score. We believe the result benefits from the fact that the model captures some significant information for classifying unknown classes. Thus, we believe our model works in a reasonable way to help classify unseen classes under GZSL setting. We also plot the confusion matrix for our best model in terms of GZSL metrics in Figure 5. The first 150 classes are seen classes and the last 50 are unseen classes. We can see a highlighted diagonal in the figure, which means that most of classes are predicted correct in most cases. Nevertheless, We see a fading of colour on the diagonal for last 50 classes, which shows that the accuracy on unseen classes is much lower than that on seen classes. The figure shows the gap between the performance on seen classes and that on unseen classes, which is yet to be narrowed.

5. Conclusion and Future Work

In general, the models we propose achieve higher classification accuracy on unseen classes, leading to a better GZSL-H score compared to most of previous methods. Among all the approaches we proposed, Multi-Attention Model performs the best in terms of GZSL-H.

Note that in our approach of calibration with confidence, the confidence vector can be viewed as a conditional representation. Thus, the idea of feature-wise transformations(Perez et al., 2018) can be applied here. For the future work, we will incorporate Feature-wise Linear Modulation (FiLM)(Perez et al., 2018) into Relation Module. In addition, we maintain a single Relation Module for all classes (seen and unseen) currently. We want to adapt the concept of Parameter Generation (Platanios et al., 2018) here and create a generalized parameter generator to generate parameters for classifier of each class given class embeddings.

References

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

Bahdanau, D., Cho, K., and Bengio, Y. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

Changpinyo, S., Chao, W.-L., Gong, B., and Sha, F. Synthesized classifiers for zero-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5327–5336, 2016.

Chen, L.-C., Yang, Y., Wang, J., Xu, W., and Yuille, A. L. Attention to scale: Scale-aware semantic image segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3640–3649, 2016.

Frome, A., Corrado, G. S., Shlens, J., Bengio, S., Dean, J., Mikolov, T., et al. Devise: A deep visual-semantic embedding model. In *Advances in neural information processing systems*, pp. 2121–2129, 2013.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Huang, G., Liu, Z., Van Der Maaten, L., and Weinberger, K. Q. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4700–4708, 2017.

Kodirov, E., Xiang, T., and Gong, S. Semantic autoencoder for zero-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3174–3183, 2017.

Lampert, C. H., Nickisch, H., and Harmeling, S. Learning to detect unseen object classes by between-class attribute transfer. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 951–958. IEEE, 2009.

Norouzi, M., Mikolov, T., Bengio, S., Singer, Y., Shlens, J., Frome, A., Corrado, G. S., and Dean, J. Zero-

shot learning by convex combination of semantic embeddings. *arXiv preprint arXiv:1312.5650*, 2013.

Perez, E., Strub, F., De Vries, H., Dumoulin, V., and Courville, A. Film: Visual reasoning with a general conditioning layer. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

Platanios, E. A., Sachan, M., Neubig, G., and Mitchell, T. Contextual parameter generation for universal neural machine translation. *arXiv preprint arXiv:1808.08493*, 2018.

Reed, S., Akata, Z., Lee, H., and Schiele, B. Learning deep representations of fine-grained visual descriptions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 49–58, 2016.

Robbins, H. and Monro, S. A stochastic approximation method. *The annals of mathematical statistics*, pp. 400–407, 1951.

Romera-Paredes, B. and Torr, P. An embarrassingly simple approach to zero-shot learning. In *International Conference on Machine Learning*, pp. 2152–2161, 2015.

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

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

Sung, F., Yang, Y., Zhang, L., Xiang, T., Torr, P. H., and Hospedales, T. M. Learning to compare: Relation network for few-shot learning. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Jun 2018a. doi: 10.1109/cvpr.2018.00131. URL http://dx.doi.org/10.1109/CVPR.2018.00131.

Sung, F., Yang, Y., Zhang, L., Xiang, T., Torr, P. H., and Hospedales, T. M. Learning to compare: Relation network for few-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1199–1208, 2018b.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. In *Advances in neural information processing systems*, pp. 5998–6008, 2017.

Wah, C., Branson, S., Welinder, P., Perona, P., and Belongie, S. The caltech-ucsd birds-200-2011 dataset. 2011.

Wang, P., Liu, L., Shen, C., Huang, Z., van den Hengel, A., and Tao Shen, H. Multi-attention network for one shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2721–2729, 2017.

Xian, Y., Lampert, C. H., Schiele, B., and Akata, Z. Zero-shot learning-a comprehensive evaluation of the good, the bad and the ugly. *IEEE transactions on pattern analysis and machine intelligence*, 2018a.

Xian, Y., Lorenz, T., Schiele, B., and Akata, Z. Feature generating networks for zero-shot learning. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Jun 2018b. doi: 10.1109/cvpr.2018.00581. URL http://dx.doi.org/10.1109/CVPR.2018.00581.

Xiao, T., Xu, Y., Yang, K., Zhang, J., Peng, Y., and Zhang, Z. The application of two-level attention models in deep convolutional neural network for fine-grained image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 842–850, 2015.

Zhang, L., Xiang, T., and Gong, S. Learning a deep embedding model for zero-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2021–2030, 2017.