2023 International Conference on Pattern Recognition, Machine Vision and Intelligent Algorithms (PRMVIA) | 979-8-3503-4659-6/23/$31.00 ©2023 IEEE | DOI: 10.1109/PRMVIA58252.2023.00010

Zero-Shot Learning based on Vision Transformer

1st Ruisheng Ran*
School of Computer and Information Science
Chongqing Normal University
Chongqing, China
rshran@cqnu.edu.cn

2nd Qianwei Hu
School of Computer and Information Science
Chongqing Normal University
Chongqing, China
huqianv@163.com

3rd Tianyu Gao
School of Computer and Information Science
Chongqing Normal University
Chongqing, China
kaotienyu@163.com

4th Shuhong Dong
School of Computer and Information Science
Chongqing Normal University
Chongqing, China
domshuhom@163.com

***Abstract*—Zero-Shot Learning (ZSL) simulates human's transfer learning mechanism, which can recognize samples or categories that have not appeared during the training phase. However, the current ZSL still has a domain shift issue. To solved the domain shift issue, we propose a new ZSL method that combines Vision Transformer (ViT) and the encoder-decoder mechanism. This method refers to ViT's Multi-Head Self-Attention (MSA) to extract more detailed visual features. The encoder-decoder mechanism can make the semantic information extracted from the image features accurately express its visual features and enhance recognition accuracy. We implemented it on three data sets of CUB, SUN and AWA2, and the experimental results proved that the method suggested in this study performs better than the current available methods. It shows that our new method is an effective ZSL method.**

Keywords-Zero-shot learning; Vision Transformer; Visual features

I. Introduction

In real life, some objects on the earth are now endangered, and there are few or no samples available. Traditional classification methods cannot identify classes that didn't appear during the training phase. When new classes appear and data are constantly added, the model must be retrained, which will inevitably increase the training cost. Therefore, zero-shot learning (ZSL) emerges [1], which trains visible classes to recognize invisible classes.

At present, we divide ZSL methods into the following learning mechanisms, including the direct and indirect learning of the classifier of each attribute for prediction [2], Learn to project the visual feature space to the semantic space for model learning and then predict through the trained model [3, 4, 5, 6, 7, 8, 9, 10, 11], Simultaneously project visual features and semantic information into a shared latent space, learn a function or model for each class, and then predict the class [12]. And generative models [13]. However, many methods still encounter the domain shift issue. This is because the same attribute can exist in various categories, but the corresponding visual features may differ significantly. Especially, the visual features of the visible and invisible classes are distinct, but they have the same attributes. Thus, When a model that has been trained on visible classes is used to test invisible classes, there will be a significant deviation.

Vision Transformer (ViT) [14] was proposed by Google in 2020 to use Transformer in computer vision. The ViT model can capture global and local features and can extract more sufficient and even more subtle visual feature representations.

We suggest a novel ZSL method based on ViT and the encoder-decoder mechanism to facilitate addressing the domain shift issue. Its major contributions include:

1) This method simulates the encoder-decoder model, the encoder maps images's visual features to the semantic attributes space, and the decoder reconstructs semantic attributes into visual features of the original images. As a constraint, the decoder effectively reconstructs the visual features of the visible and invisible classes, making the trained model less susceptible to domain shift and can be more effectively applicable to Generalized Zero-Shot Learning (GZSL).

2) This method refers to ViT's Multi-Head Self-Attention (MSA) block to extract more detailed visual features so that this method can learn a model closer to semantic features and more accurately predict invisible classes, and improve the recognition accuracy of ZSL.

II. Methodology

A. Model Framework

Figure 1 is a novel ZSL method proposed by us based on ViT model and encoder-decoder mechanism. In training phase, the working principle of the encoder-decoder part is similar to that of the ViT. The sample of the visible classes set $S_{seen} = \{x, y \mid x \in X^{seen}, y \in Y^{seen}\}$ is made as input images, where X^{seen} are the visible classes sample set, and Y^{seen} is a set of class-level attribute vectors annotating the visible classes with M different attributes. First, the sample image x is converted into a resolution of 224 × 224 RGB image with 3 channels, i.e., $x \in R^{224\times224\times3}$. Then it is divided into 196 patches, each patch with a resolution of 16×16. They are flattened into a patch sequence, and then with a trainable 2D convolution layer, patch embeddings are used as output. Then, to maintain the relative positional information between image patches, position embeddings

979-8-3503-4659-6/22/$31.00 ©2022 IEEE
DOI 10.1109/PRMVIA58252.2023.00010

are introduced and added to the patch embeddings. A learnable classification token is also added at the beginning of the image patch embedding. Finally, the patch embedding is projected to 768 dimensions.

The patch embedding is then fed into the encoder consisting of MSA and MLP blocks, where Layer Norm is used before each block input, the residual connections are used after each block output, and visual features are finally output. In MSA block, the attention matrix of the image patch is obtained according to the Multi-head self-attention mechanism. Multiple attention matrices are then concatenated into the MLP module consisting of two linear projections.

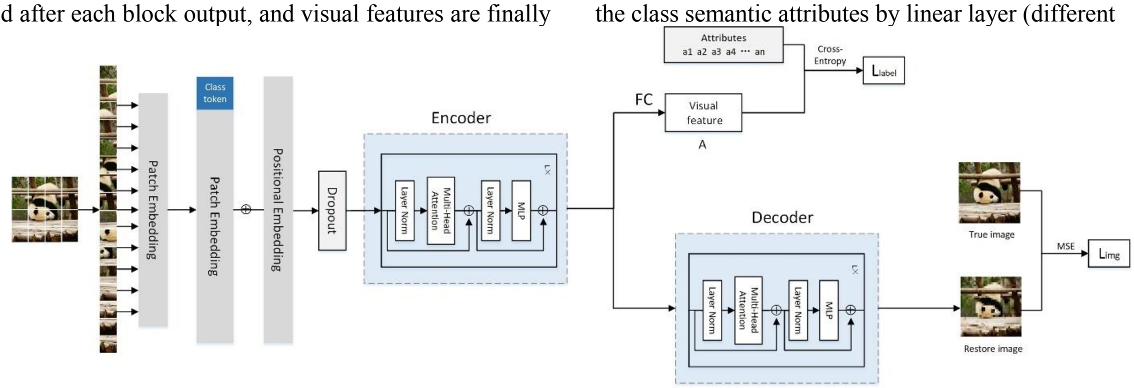

Figure 1. The architecture of our method.

Then, using its visual features, the classification header realized by the MLP module is extracted and projected to the class semantic attributes by linear layer (different datasets have different dimensions of semantic attributes) to obtain the projected visual features $f_s(x)$. The class logits $V_{attribute}(y)$ are then calculated by the dot product between each class semantic attribute (such as the color of a panda, and the stripes of a tiger, the tail of a monkey) and visual features. Next, the attribute cross-entropy loss L_{label} is employed to encourage the visual features of samples to be more similar to their corresponding semantic attributes, and the model parameters are optimized, as shown in Eq. (1):

$$L_{label} = -\frac{1}{m}\sum_{i=1}^{m}\log\frac{\exp(f_s(x_i)V_{attribute}(y)^T)}{\sum_{\hat{y}\in Y^s}\exp(f_s(x_i)V_{attribute}(\hat{y})^T)} \quad (1)$$

Here, m refers to the batch size of the training samples $\{x_i\}_{i=1}^{m}$.

In this method, the decoder and encoder models are structurally symmetric, which is a constraint. To make the features extracted by the encoder more accurate, the decoder is also composed of the MSA and MLP blocks. In this part, the encoder's output is utilized as the input to the decoder for the purpose of image reconstruction. Let x_i are the input original image and $\hat{x}_i$ are the output reconstructed image. What we aim for is to reduce the discrepancy or difference between the output reconstructed image and the input original image to the smallest possible value, so we express this goal with the MSE (Mean Square Error) and call it the image loss L_{img}, as in Eq. (2),

$$L_{img} = \frac{1}{m}\sum_{i=1}^{m}(\hat{x}_i - x_i)^2 \quad (2)$$

Then, the label loss and the image loss are taken as the total loss L, as in Eq. (3), where λ is a hyperparameter, and the total loss is backpropagated to train the whole model.

$$L = L_{label} + \lambda L_{img} \quad (3)$$

L_{img} is similar to the "Regressor" module in GDAN [15] and FrWGAN [16], but there are many differences. The concept of L_{img} is derived from the principles of Autoencoder [17], which utilizes the encoder-decoder mechanism. The idea of the "Regressor" in GDAN and FrWGAN is from the "cycle consistency loss" [18]. L_{img} is based on the Transformer, especially, Multi-Head Self-Attention (MSA) block to extract more detailed visual features. The Regressor module is a multilayer perceptron in FrWGAN, or feed-forward neural networks in GDAN. The L_{img} is used to represent the semantic → visual method, but the Regressor represents the visual → semantic method.

B. invisible classes prediction

ZSL: in this part, The training samples and test samples are separate or not overlapping. In testing phase, only the test samples belonging to the set of invisible classes $S_{unseen} = \{x, y \mid x \in X^{unseen}, y \in Y^{unseen}\}$ are used, where X^{unseen} are the sample sets, and Y^{unseen} are the class-level attribute vector sets. These samples are firstly pre-processed in the same way as the training phase. Then entering the trained encoder, the visual features of the test sample images are extracted, and trained linear layers are used to map these visual features to the semantic attribute space, resulting in the obtained visual features $f_{unseen}(x)$. The class predictions are generated by taking the dot product of the visual features and semantic attributes $V_{attribute}(y)$, as in Eq. (4):

$$\hat{y} = \arg\max_{y\in Y^{unseen}} f_{unseen}(x)V_{attribute}(y)^T \quad (4)$$

GZSL: in this part, the trained model is applied to make predictions on the category or classification of the samples, which can belong to either the visible or the invisible classes set $S_{seen} \cup S_{unseen}$ in testing phase. The data used during the training phase may come from visible classes, the prediction results during the test stage could exhibit a bias towards the visible classes. To address the aforementioned issue, we employ the Calibration Stack

(CS) method [19] , which reduces the impact of visible classes by a constant factor. The prediction category formula of GZSL, as in Eq. (5):

$$\hat{y} = \underset{y \in Y^{unseen} \bigcup Y^{seen}}{\arg\max} f_{unseen \bigcup seen}(x) V_{attribute}(y)^{T} - \xi II[y \in Y^{seen}] \quad (5)$$

Here, if y is a visible class, then $II[\bullet]=1$, and 0 otherwise. ξ denotes the calibration factor.

III. Experiments and Results

Our proposed method uses the pre-trained model VIT-base [14], which has 768 hidden dimensions, 12 heads per layer, an MLP size of 3072, and 86 million parameters of this model. For AWA2 and CUB datasets, the CS factor ξ is 0.9, while for SUN dataset, it is 0.4. The above values of ξ are empirical. Experiments were implemented on three datasets, and the ZSL and GZSL results indicate the efficacy of our method.

A. Datasets

The datasets used in this study are classic public datasets, which include AWA2 [20], CUB [21], and SUN [22]. To ensure that the pre-trained model does not have any test classes in its training set, a careful selection of the training and testing datasets is conducted, All three datasets follow the Split (PS) data partitioning proposed in [20]. The data division is shown in Table Ⅰ.

TABLE I. Experimental datasets

Datasets	**SA**	**tr**	**ts**	**Images (total)**	**training stage**	**testing stage**	
					Train+val	***S***	***U***
SUN	102	580+75	72	14340	10320	2580	1440
CUB	312	100+50	50	11788	7057	1764	2967
AWA2	85	27+13	10	37322	23527	5882	7913

The number of classes during training and validation (tr) and testing (ts), the total number of images during training and validation (train+val), visible (S) and invisible (U) testing, and the dimensionality of semantic attributes (SA).

B. Evaluation

For ZSL, the accuracy rate of top-1 Acc_{y} as the evaluation standard as in Eq. (6).

$$Acc_{y} = \frac{1}{\|Y\|} \sum_{C=1}^{\|Y\|} \frac{n_{C}}{N_{C}} \quad (6)$$

Here, n_{C} is the all number of samples from a specific class C that are predicted correctly, and N_{C} is the all number of samples belonging to class C.

For GZSL, we introduce top-1 accuracy, In order to calculate the accuracy of both visible and invisible classes denoted as $Acc_{y^{s}}$ and $Acc_{y^{u}}$, respectively, and use the harmonic average as the evaluation standard, as in Eq. (7).

$$H = \frac{2 \times Acc_{y^{u}} \times Acc_{y^{s}}}{Acc_{y^{u}} + Acc_{y^{s}}} \quad (7)$$

C. The Comparison of Results

Experiments are carried out on the above datasets, and the method is compared with several other advanced ZSL and GZSL, as shown in Tables Ⅱ and Ⅲ, where the highest data is bolded and italicized, and the evaluation of results is based on the average top-1 accuracy (%) for each class.

ZSL's results are presented in Table Ⅱ. Our method is compared with many existing methods, and their performance is evaluated, although Our method performs 2.5% worse than the SRSA method on AWA2, but outperforms CUB and SUN by 10.0% and 6.1%, respectively. The results say that our proposed method obtains more precise visual features on fine-grained datasets, and has more advantages and great potential on fine-grained datasets.

TABLE II. ZSL results

Methods	**AWA2**	**CUB**	**SUN**
DAP[2]	46.1	40.0	39.0
DeViSE[11]	59.7	52.0	56.5
ConSE[6]	44.5	34.3	38.8
SSE[12]	61.0	43.9	51.5
SJE[4]	61.9	53.9	53.7
ESZSL[7]	58.6	53.9	54.5
LATEM[23]	55.8	49.3	55.3
ALE[3]	62.5	54.9	58.1
SAE[8]	54.1	53.6	59.7
PSRZSL[9]	63.8	56.0	61.4
SRSA[24]	***68.3***	59.9	64.3
FrWGAN[16]	-	58.6	59.9
Our method	65.8	***69.9***	***70.4***

Our proposed method mainly focuses on GZSL, and Table Ⅲ shows the experimental results of GZSL. In this table, Se denotes accuracy of visible classes, Un denotes accuracy of invisible classes, and H denotes the harmonic average. Compared with these methods, the harmonic mean values of our method on AWA2, CUB, and SUN are at least 1.7%, 8.4%, and 4.4% higher, respectively, showing a significant improvement in our method. And CUB is at least 4.9% higher than other methods in invisible classes. From the experimental results, it can be seen that our method is better than other methods

TABLE III. GZSL result

Methods	**AWA2**			**CUB**			**SUN**		
	Se	***Un***	***H***	***Se***	***Un***	***H***	***Se***	***Un***	***H***
DAP[2]	84.7	0.0	0.0	67.9	1.7	3.3	25.1	4.2	7.2
DeViSE[11]	74.7	17.1	27.8	53.0	23.8	32.8	30.5	14.7	19.8

ConSE[6]	***90.6***	0.5	1.0	72.2	1.6	3.1	39.9	6.8	11.6
SSE[12]	82.5	8.1	14.8	46.9	8.5	14.4	36.4	2.1	4.0
SJE[4]	73.9	8.0	14.4	59.2	23.5	33.6	30.5	14.7	19.8
ESZSL[7]	77.8	5.9	11.0	63.8	12.6	21.0	27.9	11.0	15.8
LATEM[23]	77.3	11.5	20.0	57.3	15.2	24.0	28.8	14.7	19.5
ALE[3]	81.8	14.0	23.9	62.8	23.7	34.4	33.1	21.8	26.3
SAE[8]	82.2	1.1	2.2	54.0	7.8	13.6	18.0	8.8	11.8
SGMA[25]	87.1	37.6	52.5	71.3	36.7	48.5	-	-	-
GAZSL[26]	86.5	19.2	31.4	60.6	23.9	34.3	34.5	21.7	26.7
DVN[5]	73.4	34.9	48.5	55.1	26.2	35.5	34.6	25.3	29.2
PSRZSL[9]	73.8	20.7	32.3	54.3	24.6	33.9	37.2	20.8	26.7
DDSA[10]	82.8	28.7	42.6	53.9	25.1	34.3	33.9	22.3	26.9
E-PGN[27]	83.5	***52.6***	64.6	61.1	52.0	56.2	-	-	-
ACGN[28]	87.5	20.8	33.6	57.4	20.9	30.6	33.8	16.5	22.1
SRSA[24]	59.6	38.1	46.5	55.6	27.5	36.8	37.9	25.3	30.3
FrWGAN[16]	-	-	-	59.3	47.9	53.0	33.8	***47.2***	39.4
Our method	90.1	52.4	***66.3***	***74.8***	***56.9***	***64.6***	***52.8***	37.4	***43.8***

D. Hyperparameter Analysis

A hyperparameter λ is incorporated in this method to fine-tune the balance between the label loss and the image loss, aiming to train an optimal network model. In the experiment, we evaluate the interval [0.001, 0.1] on those datasets. The results depicted in Fig. 2 (a), (b), (c), and (d) illustrate the efficacy of our method across the three datasets and the harmonic mean of accuracy. When λ taking different values, it can be visible that the accuracy is relatively low when λ is large or small. Therefore, when λ is set to 0.01, this method achieves the best performance.

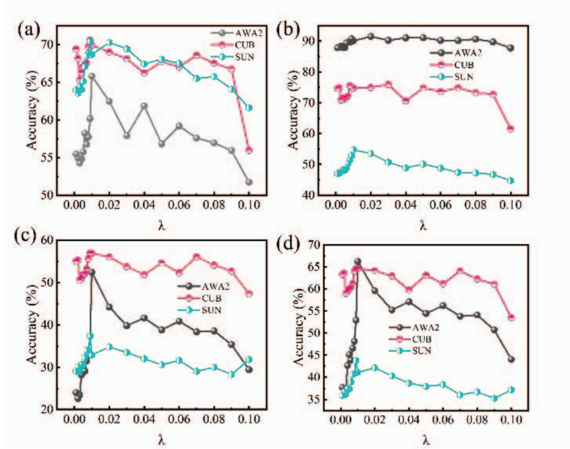

Figure 2. Hyperparameter analysis. (a) accuracy of ZSL, (b) accuracy of GZSL's visible classes, (c) accuracy of GZSL's invisible classes (d) the harmonic average of GZSL.

E. Ablation Experiment

In this part, the effectiveness of calibrated stacking (CS) is presented in Table 4. The baseline is a method without CS. Baseline + CS means CS is implemented on all test sets. As shown in this table, it can be visible that CS effectively solves the problem that the prediction results in the test stage will be biased towards the visible classes.

TABLE IV. ABLATION PERFORMANCE OF OUR METHOD ON GZSL

Methods	**AWA2**			**CUB**			**SUN**		
	Se	***Un***	***H***	***Se***	***Un***	***H***	***Se***	***Un***	***H***
Baseline	95.9	22.1	35.9	86.2	29.9	44.4	58.3	29.7	39.4
Baseline+CS	90.1	52.4	66.3	74.8	56.9	64.6	52.8	37.4	43.8

IV. CONCLUSIONS

This paper suggests a novel ZSL approach using ViT and the encoder-decoder mechanism. This method alleviates the domain shift problem. Our method has been demonstrated to outperform traditional and contemporary ZSL and GZSL techniques, as evidenced by experimental results on three datasets. In the future, we can consider improving the structure of the Transformer to get better recognition accuracy.

REFERENCES

[1] M. Palatucci, D. Pomerleau, G. E. Hinton, and T. M. Mitchell, "Zero-shot learning with semantic output codes," *Advances in neural information processing systems,* vol. 22, 2009.

[2] C. H. Lampert, H. Nickisch, and S. Harmeling, "Learning to detect unseen object classes by between-class attribute transfer," *2009 IEEE conference on computer vision and pattern recognition*: IEEE, pp. 951-958, 2009.

[3] Z. Akata, F. Perronnin, Z. Harchaoui, and C. Schmid, "Label-Embedding for Image Classification," IEEE Transactions on Pattern Analysis & Machine Intelligence, vol. 38, no. 7, pp. 1425-1438, 2016.

[4] Z. Akata, S. Reed, D. Walter, H. Lee, and B. Schiele, "Evaluation of output embeddings for fine-grained image classification," Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 2927-2936, 2015.

[5] H. Zhang, L. Yang, W. Yang, and S. Ling, "Dual-Verification Network for Zero-shot Learning," Information sciences, vol. 470, pp. 43-57, 2018.

[6] M. Norouzi *et al.*, "Zero-shot learning by convex combination of semantic embeddings," *arXiv preprint arXiv:1312.5650,* 2013.

[7] B. Romera-Paredes and P. Torr, "An embarrassingly simple approach to zero-shot learning," *International conference on machine learning*: PMLR, pp. 2152-2161, 2015.

[8] E. Kodirov, T. Xiang, and S. Gong, "Semantic autoencoder for zero-shot learning," *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3174-3183, 2017.

[9] Y. Annadani and S. Biswas, "Preserving semantic relations for zero-shot learning," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7603-7612, 2018.

[10] Y. Liu, J. Li, and X. Gao, "A simple discriminative dual semantic auto-encoder for zero-shot classification," *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pp. 940-941, 2020.

[11] A. Frome *et al.*, "Devise: A deep visual-semantic embedding model," *Advances in neural information processing systems,* vol. 26, 2013.

[12] Z. Zhang and V. Saligrama, "Zero-shot learning via semantic similarity embedding," *Proceedings of the IEEE international conference on computer vision*, pp. 4166-4174, 2015.

[13] Y. Zhu, M. Elhoseiny, B. Liu, and A. Elgammal, "Imagine it for me: Generative Adversarial Approach for Zero-Shot Learning from Noisy Texts," 2017.

[14] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, and N. Houlsby, "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale," *arXiv preprint arXiv:2010.11929,* 2020.

[15] H. Huang, C. Wang, P. S. Yu, and C.-D. Wang, "Generative dual adversarial network for generalized zero-shot learning," Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 801-810, 2019.

[16] R. Felix, I. Reid, and G. Carneiro, "Multi-modal cycle-consistent generalized zero-shot learning," *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 21-37, 2018.

[17] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," science, vol. 313, no. 5786, pp. 504-507, 2006.

[18] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," Proceedings of the IEEE international conference on computer vision, pp. 2223-2232, 2017.

[19] 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," European conference on computer vision: Springer, pp. 52-68, 2016.

[20] 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, vol. 41, no. 9, pp. 2251-2265, 2018, doi: 10.1109/TPAMI.2018.2857768.

[21] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, "The caltech-ucsd birds-200-2011 dataset," california institute of technology, 2011.

[22] G. Patterson and J. Hays, "Sun attribute database: Discovering, annotating, and recognizing scene attributes," *2012 IEEE Conference on Computer Vision and Pattern Recognition*: IEEE, 2012.

[23] Y. Xian, Z. Akata, G. Sharma, Q. Nguyen, M. Hein, and B. Schiele, "Latent embeddings for zero-shot classification," Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 69-77, 2016.

[24] Y. Liu, X. Gao, J. Han, L. Liu, and L. Shao, "Zero-shot learning via a specific rank-controlled semantic autoencoder," Pattern Recognition, vol. 122, p. 108237, 2022.

[25] Y. Zhu, J. Xie, Z. Tang, X. Peng, and A. Elgammal, "Semantic-guided multi-attention localization for zero-shot learning," *Advances in Neural Information Processing Systems,* vol. 32, 2019.

[26] Y. Zhu, M. Elhoseiny, B. Liu, X. Peng, and A. Elgammal, "A generative adversarial approach for zero-shot learning from noisy texts," Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1004-1013, 2018.

[27] Y. Yu, Z. Ji, J. Han, and Z. Zhang, "Episode-based prototype generating network for zero-shot learning," Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 14035-14044, 2020.

[28] J. Liu, Z. Zhang, and G. Yang, "Cross-class generative network for zero-shot learning," Information Sciences, vol. 555, pp. 147-163, 2021.