EVOLUTION OF CLASS EMBEDDING FOR UNSEEN CLASS IN ZERO-SHOT LEARNING

[1]SHENG-TZONG CHENG, [2]CHUNG-CHIH PAN

[1,2]Computer Science and Information Engineering Department, National Cheng Kung University, Tainan, Taiwan
E-mail: [1]stevecheng1688@gmail.com,[2]pansam9961217@gmail.com

Abstract - In this paper, we propose a novel method to improve the generation of the CE with the following two characteristics: (1) It can quantify the value of feature. and (2) In addition to utilizing the "positive" features, our method takes into account the impact of the "negative" features. We name our method as P-learning model. In this paper, we conduct experiments to replace the CE learned from the P-Learning model with the CE provided in the data set with the ZSL Model designed by others. Under the three different accuracy levels of Seen / Unseen / Generalized, the accuracy by our method is much higher than that reported in other papers. It shows that the CE learned by the P-Learning Model is very helpful and feasible in classification.

Keywords - Zero-shot Learning, Class Embedding, Variational Auto-Encoder, Knowledge Representation.

I. INTRODUCTION

In recent years, machine learning has been widely used in various fields, such as Alpha go defeating the human chess king, Tesla using image recognition technology to achieve autonomous driving, and Apple using natural language technology to improve applications such as the voice assistant Siri. The speed of machine learning technology development is far beyond our imagination, but behind such a powerful technology, it is supported by many training materials. In practice, when we plan to apply machine learning algorithms in the real world, we will face many problems, such as domain adaptation and knowledge representation in transfer learning. To tackle the bottlenecks, Zero-Shot Learning (ZSL) [1, 4, 7, 8] was proposed to solve the difficulty in collecting training data. This technology uses a semantic description vector or class embedding (CE) to describe seen and unseen classes. The known knowledge of the seen class is transferred to the unseen class to identify objects of the unseen class.ZSL might be short of applying in practice. Therefore, Generalized Zero-Shot Learning (GZSL) [5, 6, 10, 11, 12] is proposed as an augmented setting for ZSL. In GZSL, test data comes from both seen and unseen classes. GZSLis able to classify data to the corresponding category In a typical ZSL, the CE of the seen class and the unseen class must be used as input for training. There are currently two ways to generate CE, one is through expert definition (more time-consuming), and the other is through natural language technology in Wikipedia (more difficulty). Moreover, in the real world, it is difficult for us to define its class embedding for a category that has never been seen before. For example, the mating of elephants and whales produces new species, but we don't have CE for this species. As a sequence, the new species cannot be classified by the machine. We propose a novel learning technique, P-learning, to solve the problem of "generating /defining CE for categories that have never been seen before". The inspiration of this model comes from the concept of Virtual Intelligence Evolution (VIE), which outlines the methodology of the model, and evolution means the process from knowing nothing to something. As illustrated in Fig.1.,let us consider a classic example. Normally children recognize the semantic descriptions of the shape of a horse, the stripes of a tiger, and the black and white of a penguin. If there is a zebra that a child never saw before, the child recognizes the new thing according to the semantic descriptions he knows. These semantic descriptions form a new category. This human-learning example could be analog to the machine VIE, where the distinguishing capability by CEto discover the unseen category is a process of machine evolution.

One trait of the P-learning is the construction of the mapping between feature space and semantic space. The framework is built on Conditional Variational Auto-Encoder (CVAE) [2, 3, 13] and divided into two stages. The first stage is to train the Residual Network (ResNet) to extract 2048-dimensional features [9] of the picture. The second stage is to train CVAE-based encoder to learn the mapping relationship between feature space and semantic space. P-learning is able to identify Seen Class and Unseen Class through the parsed semantic vector. This paper makes the following technical contributions:

1. A new CE generation method is proposed.
2. The CE learned by the P-learning can be quantified and has its own way of explanation. Mainly, the outcome of the P-learning is explainable.
3. The CE of each species not only shows "positive" characteristics but also has "negative" characteristics.
4. High accuracy of data tests is achieved by using the CE learned by the P-Learning. Therefore, the

CE by the P-learning could replace the CE defined by the human experts.

5. By applying the P-learning methodology, we compile a new ZSL data set, called the Plant Diseases Data Set, with 19,000 photos and 46-dimensional CE. This is the first attempt of providing CE for Plant Disease Diagnosis in the literature.

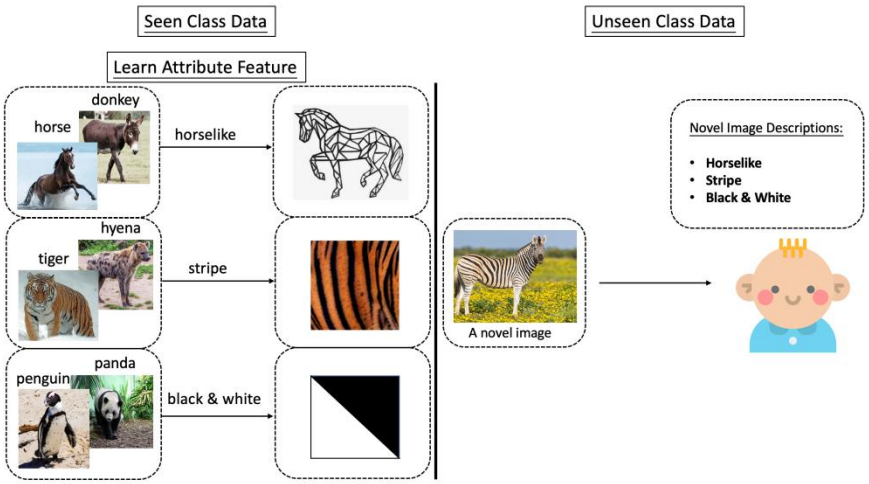


Fig.1. The knowledge of the seen class transferred to the unseen class

II.OUR APPROACH

2.1. Architecture of P-learning

In ZSL and GZSL, the CE of the classes needs to be provided before training. In other words, if the CE of a particular unseen class is unknown yet at the training time of the model, we will not be able to use ZSL or GZSL to deal with the unseen classes. As we know, in the particular area of recognizing living creatures, there are still many animals in nature that have not been discovered by humans. Under such context, we do not have information on these species at hand. If we hope that a learning model can identify these unseen classes and ask experts to spend time in defining the CE for the unseen classes, it would be impractical. This motivates the development of P-learning.

P-learning is a method to solve the problem of "defining CE for categories that have never been seen before". Given an unseen domain category, P-learning is able to generate the CE for it by exploiting the CE of the seen classes. The construction of the CE forms a basis for explainable learning mechanism. For different domains, the dimensions of the CE will be different. For example, 85 dimensions of the CE are associated with the AWA2 [14] data set, and 312 dimensions are with CUB [15]. In this work, we contribute to provide a data set for plant [17], which has 46 dimensions of CE.

The system architecture of P-learning, shown in Fig.2., can be depicted as two stages. In the first stage, the feature extraction model obtains the basic elements that appear in the domain. ResNet could be a perfect backbone for the feature extraction model by pretraining. The reason why we choose the ResNet is because the ResNet model stands out in the ImageNet competition and has a very powerful generalization capability in recognition tasks. However, Taking the AWA2 dataset as an example, the 2048-dimensional features are basically only machine-readable. From a human perspective, it is not exactly the way how a human recognizes an animal. A human recognizes an animal by her furry skin, sharp teeth, body colors... etc. The human perspective forms the structure of the CE. Therefore, in the second stage, through the constraints of the VAE and the proposed loss functions, the mapping relationship between the feature space (i.e. the "X" shown in Fig.2.) and the semantic space (i.e. CE) is established. The encoder is used to comprehend the attributes [16] and values of the unseen class.

P-learning proceeds as follows. The first stage is the generation of 2048-dimension machine readable features for each class Each image from an identical seen class will be fed in ResNet 101 to generate its corresponding 2048-dimension machine-readable feature. After all training images from each seen class have been processed, the output will be a set of 2048-dimension features (indicated as X in Fig.2.) for each seen class. Then, in the second stage, these features collected by each class will be fed in to train the proposed encoder and decoder. CE of each seen class from the original dataset is used to compute the loss functions. The value computed by the loss functions are used to train the encoder and decoder. As a result, the encoder after training is able to generate CE for the unseen class. Notably, the proposed encoder uses the machine-readable features established in the first stage as the input to encode the CE of the domain. The relationship between the seen and unseen classes is embedded in CE. The relationship includes the similarity and the difference between the cluster of seen classes and the cluster of unseen classes. The

two clusters share a certain length of dimensions. For example, the animal domain in the AWA2 dataset uses 85 dimensions to depict, including color, animal body type, background, exterior, and so on.

To make VAE work, two types of loss function were proposed, namely the reconstructed loss function and the CE-constrained loss function. The purpose of the reconstructed loss function is to converge the gap between image feature and output that is reconstructed by decoder. The purpose of the CE-constrained loss function is to allow the CE generated by the encoder to align with the CE provided by the data set. In addition, we confine the CE fitted by the model to conform to the normal distribution. Namely, for each category's distribution of the CE, the value of mean equals zero and the value of the variance equals one.

2.2. Training Residual Network

As mentioned in Section A, the ResNet is used as a tool for extracting features. To make the extracted features work, we focus on the validation accuracy of classification task for the seen class. In the training phase, the input of the model is the data of the seen class and the label associated with the data. Upon training, the pre-trained weights are loaded. Based on the pre-trained weights, it exhibits a strong recognition ability and is able to greatly shorten the training time and computing resources. High validation accuracy rates can be obtained on data sets such as AWA2, CUB, and plants.

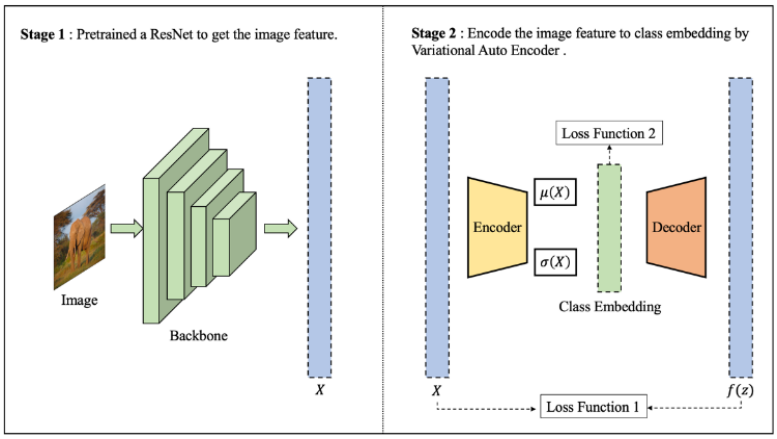


Fig.2. The Proposed Architecture

In the choice of the optimizer, "SGD + momentum" is better than Adam, and "SGD + momentum" has a chance to achieve global minima, Therefore, we choose "SGD + momentum" as the optimizer in the experiments of next chapter. After the model training is done, we remove the fully connected layer and the softmax layer. And the output of global average pooling is used as the extracted feature. The number of the output dimensions is 2048. The feature extraction model is illustrated in Fig.3.

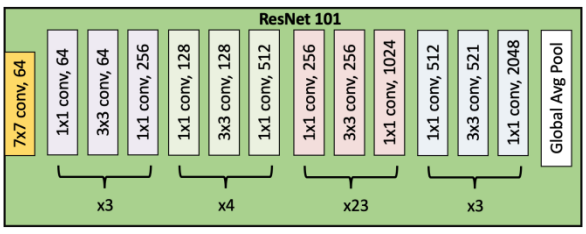


Fig.3. Feature extraction model architecture. (Remove the fully connected layer and the softmax layer from ResNet101)

2.3. Modeling CVAE

The CVAE model is adopted in our work to construct an encoder for learning the mapping relationship between machine-readable feature space and human semantic space. Upon training, the input to the proposed encoder is the 2048-dimensional feature extracted from the seen class pictures along with the description of the CEfor each seen class. The outputs of the encoder are the values for each dimension of the CE fitted by the model. According to the characteristics of the CVAE model, the encoder comply to maintain a corresponding variance (to be the value of one) for each dimension of the CE when fitting each class. Take the fitting of the class of "elephants" as an example. When predicting an elephant, the model can have a higher value in a basic attribute such as long nose, four feet, gray, big ears, and appearing on the grassland, and so on. In addition, the output fitted by the model will also meet the condition of variance equals one. The reason why the variance is maintained as one is to make sure the encoder accepts the various pictures for encoding. Another example is that the pictures for dog will have various background of a grassland, road, and home, but the encoder will be able to encode the CE for the dog category (of different background) with the value close to the mean.In summary, by feeding in seen classes, the proposed encoder will generate the corresponding CE values for each class.It satisfies the condition that variance equals one. Also, the CE precisely reflects the domain knowledge for each class.

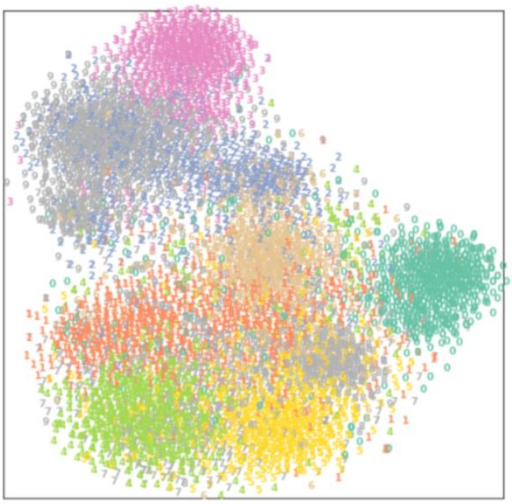


Fig.4. Ten unseen classes are encoded by the encoder. (For simplicity of viewing, the CE is projected to a two-dimensional map. Different colors denote the clustering of the unseen classes)

To meet the above conditions, two loss functions for the model are proposed. One of them comes from the modification of the Kullback-Leibler (KL) loss function. We rename it as the CE−constrained loss function. This function keeps the mean that the model fitted close to the CE provided by the dataset.

The virtue behind the KL loss function is that the concept of entropy is used to measure the error of the two probability distributions. It is able to constrain the mean and variance encoded by the encoder to satisfy the normal distribution. The following shows the derivation process for the formula.

$$
\begin{aligned}
&\mathrm{KL}(\,\mathrm{N}(\mu,\sigma^2)\ ,\ \mathrm{N}(0,1)\,)\\
&=\int \frac{1}{\sqrt{2\pi\sigma^2}}\mathrm{e}^{-\frac{(\mathrm{x}-\mu)^2}{2\sigma^2}}\left(\log\frac{\mathrm{e}^{-(\mathrm{x}-\mu)^2/2\sigma^2}/\sqrt{2\pi\sigma^2}}{\mathrm{e}^{-\mathrm{x}^2/2}/\sqrt{2\pi}}\right)\mathrm{dx}\\
&=\int \frac{1}{\sqrt{2\pi\sigma^2}}\mathrm{e}^{-(\mathrm{x}-\mu)^2/2\sigma^2}\log\left\{\frac{1}{\sqrt{\sigma^2}}\exp\left\{\frac{1}{2}[\mathrm{x}^2-(\mathrm{x}-\mathrm{u})^2/\sigma^2]\right\}\right\}\ \mathrm{dx}\\
&=\frac{1}{2}\int \frac{1}{\sqrt{2\pi\sigma^2}}\mathrm{e}^{-(\mathrm{x}-\mu)^2/2\sigma^2}\left[-\log\sigma^2+\mathrm{x}^2-\frac{(\mathrm{x}-\mu)^2}{\sigma^2}\right]\mathrm{dx}\\
&=\frac{1}{2}[-\log\sigma^2+\ \mu^2+\sigma^2-1]
\end{aligned}
$$

To constrain the CE encoded by the encoder, the real mean value of CE is subtracted from the computed mean. The CE−constrained function shown in Eq. (1) is exploited to make sure that the CE meets the condition of normal distribution by CVAE, that is, mean equals zero and variance equals one. The CE-constrained function corresponds to the Loss function 2 shown in Fig.2.

In addition, Binary Cross Entropy (BCE) function is used to measure the error between the original and the reconstructed data through the decoder. Therefore, the other function used in our model is the Reconstruction function shown in Eq. (2). Upon training the model, the input to the encoder and the output of the decoder are fed in to compute the error of reconstruction. The Reconstruction function corresponds to the Loss function 1 shown in Fig.2.

$$
\begin{aligned}
&CE-constrained\ function:=\\
&\quad -0.5*\ [1+log\sigma^2+\ \sigma^2+(\mu\ -\ REAL_{CE})^2]
\end{aligned}
\tag{1}
$$

$$
\begin{aligned}
&Reconstruction\ :=\\
&\quad 0.5*BCE\left(x_{Inpu\,t_{(i)}},x_{Outpu\,t_{(i)}}\right)
\end{aligned}
\tag{2}
$$

These two loss functions play significant roles in the model training process. By minimizing the value of the reconstruction error, the model tries to make the variance of the encoder fitting close to zero. On the other hand, by constraining the CE and meeting the condition of normal distribution, that is, variance equals one. As a result, a generative / adversarial process is created. When the decoder is not optimized yet, the reconstruction error is much greater than the value of the CE−constrained. The model will appropriately reduce the variance fitted by the encoder, making the reconstruction more precise, that is, the reconstruction error is reducing. When the decoder is nearly optimized, the reconstruction error will be much smaller than the value of the CE−constrained function. As a consequence, the encoder will enlarge the fitted variance, causing reconstruction error bigger. It will make the reconstruction process difficult again. So, the decoder must find a way to improve the generative process. Such a confrontation process improves the decoder and encoder at the same time. It can be seen that our proposed approach represents one of the novel mechanisms to tackle the problem of "generating CE and optimizing / balancing encoder and decoder at the same time ".

III .IMPLEMENTATION AND EXPERIMENTS

3.1. Experimental Settings

In the experiments, we use the setting listed in Table 1 for the following three data sets:

1. Animals with Attributes 2 (AWA2), including 50 classes of wild animals, a total of 37,322 pictures, and 85-dimensional CE. Among them, we assign 40 categories as the seen classes, and the remaining 10 categories as unseen classes.
2. Caltech-UCSD Birds-200-2011 (CUB), including 200 species of wild birds, a total of 11788 photos, and 312-dimensional CE. Among them, we assign 150 categories as the seen classes, and the remaining 50 categories as unseen classes.
3. The Plant data collection downloaded from Kaggle contains 38 plant diseases and insect

pests, a total of 19,000 photos, and a custom 46-dimensional CE. Among them, we assign 25 categories as the seen classes, and the remaining 13 categories as unseen classes.

	Setting
OS	Ubuntu 16.04
Platform	Tensorflow 1.14.0 GPU / Keras 2.2.5
Program language	Python 3.6
CPU	Intel(R) Core (TM) i7-6700 CPU@ 3.40GHz
GPU	Nvidia GTX1060 6G
memory	16G

Table 1: Experiment environment

During training, only the data of the seen classes will be used, and 80% of the data is assigned as the training set and 20% as the validation set. In the feature extraction stage, we will use the image and label of the seen classes to train the ResNet. Before training, we will load the pre-trained weights to reduce the experiment time. Then the ResNet model is able to deliver good generalization capability, and learn the important elements of the domain from the pictures. The validation accuracy is observed to determine whether the model has been trained well. After the feature extraction stage is completed, the ResNet has converted the visual data of the seen classes into 2048-dimensional features as the input for training our VAE model.

Before training the VAE model, the value of each dimension in the CE provided by the original data set ranges between 0 and 100. This will cause the training time of the model bigger, or even fail to converge. To solve this problem, we perform data standardization for the CE. The mathematical formula is shown in Eq. (3).

$$Z = \frac{X - \mu}{\sigma} \sim N(0,1) \qquad (3)$$

Data standardization not only reduces the training time, but also has the following advantage. The original value of some attributes is zero, and then after data standardization, it becomes negative value. Negative value indicating a negative feature, which can help the classification task to achieve higher accuracy. The existence of negative features could enlarge the distance between classes in high-dimensional space.

Upon training our proposed model, we perform the data standardization on the original CE learned from the encoder. Through Eq. (2), the CE fitted by the encoder is constrained to be close to the real CE provided by the original data set. At the same time, through the confrontation process using Eq. (1) and Eq. (2), the encoding capability of the encoder and the decoding capability of the decoder can be enhanced.

At the validation phase, we parse the corresponding CE for the unseen class data. To verify that the CE learned from training the encoder is meaningful and logical, we calculate the difference between the learned CE and the original CE provided by the dataset. To illustrate the significance of the proposed P-learning model, the learned CE for both seen classes and unseen classes is evaluated by integrating our P-learning model into several well-known learning models reported in the literature. The models considered in the experiments are TF-VAEGAN [18], CADA-VAE [19], and EZSL [20]. Accuracy is adopted as the evaluation index in the experiments. In addition, the improvement of Domain Shift is also explained.

3.2. Experimental Results

To compare the CE learned by P-Learning with the CE provided by the original data set, we define a term diff for quantization, where #j indicates the length of the CE vector and #i indicates the number of input for the class i. For example, #j equals 85 for AWA2 and #i is 10,000 for the class of "cats". L_i^j and G_i^j represent the values of the i-th element of the j-th dimension of the CEs. Note that L indicates the CE learned from our proposed model while G indicates the CE of the original dataset. An offset value, $1e-5$, is added in Eq. (4). It is used to avoid the "divide-by-zero" event when G_i^j could be zero from the original data.

$$\text{diff} = \frac{1}{\#i} * \sum_i \left(\frac{\sum_j \left(L_i^j - G_i^j \right) / \left(G_i^j + 1e-5 \right)}{\#\, j} \right) \qquad (4)$$

For the seen class, the CE learned by P-Learning is mostly consistent with the CE provided in the original data set, and the value of diff is around 0.20 to 0.25. It means that P-Learning model indeed comprehends the relationship between the feature space and the semantic description space.

For an unseen class "Persian cat", for instance, our model learned the attribute of "black". It is justified by checking the photos of the class "Persian cat". We find that the attribute of "black" was not defined in the CE of the original dataset. Notably our model is able to handle and discover such attribute when the pictures of a black Persian cat are fed in.

Another example is from observing the attributes of the giant panda. It can be found that the two most outstanding attributes of the panda, black and white, are fitted by our P-Learning model. These observations show that the P-Learning model can not only convert the feature space to the semantic space precisely, but also has its point of view to explain the unseen class logically.

Next, we use the CE learned by P-Learning to replace the CE provided by the original data set; Experimental results are provided in Table 2, where notations S, U, H in the table denote the cases for the seen, unseen, and generalized (hybrid) classes respectively. The calculation for the accuracy is done by Eq. (5).

$$H = \frac{2 * acc_{y^{tr}} * acc_{y^{ts}}}{acc_{y^{tr}} + acc_{y^{ts}}} \qquad (5)$$

Table 2 illustrates the experimental results of the three data sets AWA2, CUB, and Plant. Validation of the accuracy is performed on the TF-VAEGAN, CADA-VAE, and EZSL models with and without integrating our P-learning model. Recall that we integrate our P-learning model into these models by replacing the original CE by the learned CE. Since the Plant data set is brand new and undefined, there is no accuracy data reported before. We can find that the CE fitted by the P-Learning model achieves higher accuracy than the original ones do under all cases of unseen, seen, and generalized categories. The significance of our model can be demonstrated.

The reason why the proposed model could achieve higher accuracy is because the model can perfectly catch the mapping between the feature space and the semantic space. In addition, the CE has been standardized to make the model converge faster. Least but not the last, the CE learned by the P-Learning model makes good use of negative features, which can increase the distance in high-dimensional space and make the performance of classification better.

	AWA2			CUB			Plant		
Model/ Class	***S***	***U***	***H***	***S***	***U***	***H***	***S***	***U***	***H***
CADA-VAE	75.0	55.8	63.9	53.5	51.6	52.4	NA	NA	NA
CADA-VAE (w/ P-Learning)	**84.4**	**62.2**	**71.6**	**59.6**	**51.9**	**55.5**	**96.2**	**56.3**	**71.1**
TF-VAEGAN	75.1	59.8	66.6	64.7	52.8	58.1	NA	NA	NA
TF-VAEGAN (w/ P-Learning)	**78.7**	**73.2**	**75.9**	**63.9**	**58.1**	**60.9**	**95.2**	**79.8**	**86.9**
EZSL	NA	NA	44.8	NA	NA	NA	NA	NA	NA
EZSL (w/ P-Learning)	**89.5**	**81.3**	**76.3**	**41.2**	**67.1**	**50.5**	**99.6**	**55.1**	**65.5**

Table 2: Comparison of accuracy for the CE provided by original data set and CE learned by P-Learning

By the proposed P-learning, the 2048-dimension machine readable feature of the new specie (unseen class) is generated in the first stage, then the feature is fed in automatically in the second stage to generate a new CE. The CE reflects the concepts and knowledge of experts defining species, and customizes the definition of CE for new species through learning of known categories. By investigating the output, we conclude that the colors, scenes, and physical features demonstrate all show positive feature values, while other colors and scenes that are absent show negative feature values. This outcome justifies the significance of the P-Learning model.

IV. CONCLUSION

In this work, we proposed the P-learning architecture. P-learning Architecture consists of two stages. The first stage is the generation of 2048-dimension machine readable features for each class by using ResNet 101. Each image from the same seen class will be fed in ResNet 101 to generate its corresponding 2048-dimension feature. The output of the first stage will be a set of features for each seen class. Then in the second stage, these features collected by each class will be fed in to train the proposed encoder and decoder. CE of each seen class from the original dataset is used to compute the loss functions. As a result, the encoder after training is able to generate CE for the unseen class. The novelty for our approach is that CE for the unseen categories can be derived and generated. By conducting experiments, we found that the generated CE for unseen classes could be different from the original CE provided by the experts in the field. In the experiment of P-Learning, we showed that the generated CE could replace the original CE and effectively improve the accuracy of the model in data testing of seen/unseen/generalized categories. Specifically, for the classic AWA2 data set, we illustrated that the accuracy for the generalized category in TF-VAEGAN has increased from 66.6% to 75.9%. This justifies the significance of the P-Learning. In the ZSL field, most scholars are committed to construct a good classifier, but less efforts were made to obtain a precise CE. The P-Learning was proposed to overcome the deficiencies in this aspect. Impact of the P-Learning architecture is identified to replace the original CE and make the ZSL model having a higher accuracy. The contribution is that the evolution of CE could be expedited and automated.

REFERENCES

[1] C. H. Lampert, H. Nickisch, and S. Harmeling, "Learning to detect unseen object classes by between-class attribute transfer," IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 951-958, doi: 10.1109/CVPR.2009.5206594.

[2] Kingma, D. P. & Welling, M. (2013). Auto-Encoding Variational Bayes (cite arxiv:1312.6114)

[3] J A. Mishra, S. K. Reddy, A. Mittal and H. A. Murthy, "A Generative Model for Zero Shot Learning Using Conditional Variational Autoencoders," IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 2269-22698, doi: 10.1109/CVPRW.2018.00294.

[4] Y. Fu, T. M. Hospedales, T. Xiang and S. Gong, "Transductive Multi-View Zero-Shot Learning," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 37, no. 11, pp. 2332-2345, 1 Nov. 2015, doi: 10.1109/TPAMI.2015.2408354.

[5] 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," ECCV, 2016.

[6] Y. Xian, B. Schiele, and Z. Akata, "Zero-shot learning-the good, the bad and the ugly," IEEE Conference on Computer Vision and Pattern Recognition, 2017.
[7] M. Palatucci, D. Pomerleau, G. E. Hinton, and T. M. Mitchell, "Zero-shot learning with semantic output codes," NIPS, 2009.
[8] R. Socher, M. Ganjoo, C. D. Manning, and A. Ng, "Zero-shot learning through cross-modal transfer," NIPS, 2013.
[9] A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth, "Describing objects by their attributes," IEEE Conference on Computer Vision and Pattern Recognition, 2009.
[10] V.KumarVerma,G.Arora,A.Mishra,and P.Rai,"General- ized zero-shot learning via synthesized examples," IEEE Conference on Computer Vision and Pattern Recognition, 2018.
[11] M. Bucher, S. Herbin, and F. Jurie, "Generating visual representations for zero-shot classification," ICCV, 2017.
[12] A. Mishra, S. Krishna Reddy, A. Mittal, and H. A. Murthy, "A generative model for zero shot learning using conditional variational autoencoders," IEEE Conference on Computer Vision and Pattern Recognition, 2018.
[13] Arash Vahdat and Jan Kautz, "NVAE: A Deep Hierarchical Variational Autoencoder," IEEE Conference on Computer Vision and Pattern Recognition, 2020.
[14] Y. Xian, C. H. Lampert, B. Schiele, and Z. Akata, "Zero- shot learning-a comprehensive evaluation of the good, the bad and the ugly," TPAMI, 2018.
[15] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011dataset. 2011.
[16] Genevieve Patterson and James Hays. "Sun attribute database: Discovering, annotating, and recognizing scene attributes," IEEE Conference on Computer Vision and Pattern Recognition, pages 2751–2758, 2012.
[17] Samir Bhattarai. New Plant Diseases Dataset. Kaggle.
[18] "Latent Embedding Feedback and Discriminative Features for Zero-Shot Classification," Narayan, Sanath and Gupta, Akshita and Khan, Fahad Shahbaz and Snoek, Cees GM and Shao, Ling , ECCV, 2020
[19] "Generalized zero-and few-shot learning via aligned variational autoencoders," Schonfeld, Edgar and Ebrahimi, Sayna and Sinha, Samarth and Darrell, Trevor and Akata, Zeynep,IEEE Conference on Computer Vision and Pattern Recognition,pages 8247—8255 ,2019
[20] Bernardino Romera-Paredes and Philip H. S. Torr. 2015. "An embarrassingly simple approach to zero-shot learning," International Conference on International Conference on Machine Learning - Volume 37 (ICML'15). JMLR.org, 2152–2161.

★★★