Leveraging Self-Distillation and Disentanglement Network to Enhance Visual-Semantic Feature Consistency in Generalized Zero-Shot Learning

Xiaoming Liu , Chen Wang , Guan Yang * , Chun hua Wang , Yang Long , Jie Liu , Zhi yuan Zhang

Posted Date: 10 April 2024

doi: 10.20944/preprints202404.0672.v1

Keywords: generalized zero-shot Learning; self-distillation, disentanglement network, visual-semantic feature consistency

Article

Leveraging Self-Distillation and Disentanglement Network to Enhance Visual-Semantic Feature Consistency in Generalized Zero-Shot Learning

Xiaoming Liu [1,2], Chen Wang [1,2], Guan Yang[1,2,*], Chunhua Wang [3], Yang Long [4], Jie Liu [5,6] and Zhiyuan Zhang [1,2]

1 School of Computer Science, Zhongyuan University of Technology, Zhenzhou 450007, China; ming616@zut.edu.cn (X.L.); 2021107267@zut.edu.cn (C.W.); 2020107218@zut.edu.cn (Z.Z.)
2 Henan Key Laboratory on Public Opinion Intelligent Analysis, Zhenzhou 450007, China
3 School of Animation Academy, Huanghuai University, Zhumadian 463000, China; wangchunhua@huanghuai.edu.cn (CH.W.)
4 Department of Computer Science, Durham University, Durham, United Kingdom, yang.long@durham.ac.uk (Y.L.)
5 China Language Intelligence Research Center, Beijing 100089, China;liujie@ncut.edu.cn (J.L.)
6 School of Information Science,North China University of Technology,Beijing 100144,China
* Correspondence: yangguan@zut.edu.cn

Abstract: Generalized zero-shot learning (GZSL) aims to simultaneously recognize both seen classes and unseen classes by training only on seen class samples and auxiliary semantic descriptions. Recent state-of-the-art methods infer unseen classes based on semantic information or synthesize unseen classes using generative models based on semantic information, all of which rely on the correct alignment of visual-semantic features. However, they often overlook the inconsistency between original visual features and semantic attributes. Additionally, due to the existence of cross-modal dataset biases, the visual features extracted and synthesized by the model may also mismatch with some semantic features, which could hinder the model from properly aligning visual-semantic features. To address this issue, this paper proposes a GZSL framework that enhances the consistency of visual-semantic features using self-distillation and disentanglement network (SDDN). The aim is to utilize self-distillation and disentanglement network to obtain semantically consistent refined visual features and non-redundant semantic features to enhance the consistency of visual-semantic features. Firstly, SDDN utilizes self-distillation technology to refine the extracted and synthesized visual features of the model. Subsequently, the visual-semantic features are then disentangled and aligned using a disentanglement network to enhance the consistency of the visual-semantic features. Finally, the consistent visual-semantic features are fused to jointly train a GZSL classifier. Extensive experiments demonstrate that the proposed method achieves more competitive results on four challenging benchmark datasets (AWA2, CUB, FLO, and SUN).

Keywords: generalized zero-shot Learning; self-distillation, disentanglement network, visual-semantic feature consistency

1. Introduction

Deep learning models typically necessitate extensive, heavily labeled data during training, incurring significant human and resource costs. The introduction of Zero-Shot Learning (ZSL) effectively mitigates this constraint of deep learning models by learning the mapping relationship from auxiliary (e.g., semantic) to visual space, facilitating the classification and recognition of unseen classes [1]. However, traditional ZSL settings are somewhat idealized as they assume that the test set solely comprises samples from seen classes, which is not reflective of real-world scenarios. Generalized Zero-Shot Learning (GZSL) introduces a more rigorous task where the test set can encompass samples from both seen and unseen classes, better aligning with practical needs.

Presently, research on GZSL primarily centers on two distinct strategies. Firstly, some researchers focus on methods grounded in Generative Adversarial Networks (GANs [17,19,20,29,31,33], which employ generative models to learn the mapping relationship from semantic attributes to visual features and subsequently synthesize visual samples of unseen classes based on semantic information.

 doi:10.20944/preprints202404.0672.v1

Secondly, other researchers concentrate on embedding-based methods [14,23–27,35], striving to embed visual samples into a shared feature space to accurately reflect the semantic similarity between different classes. Through this approach, models can conduct classification reasoning using structural information in the embedding space with minimal or zero samples.

Both of these strategies align visual-semantic features through either generative or embedding methods, tackling the challenges inherent in ZSL. However, they introduce a new challenge: they often overlook the potential inconsistency in the visual-semantic features to be aligned, as illustrated in Figure 1. Most GZSL methods overlook these inconsistent visual-semantic features and forcibly align them, potentially introducing biases in visual-semantic feature alignment and undermining the recognition of unseen classes. Moreover, current GZSL approaches frequently employ pre-trained ImageNet models for extracting GZSL visual features and training generative models to synthesize visual features of unseen classes. However, the presence of cross-modal dataset bias [19] implies that the extracted and synthesized visual features might lack refinement and could stray from the visual features necessary for ZSL tasks, thereby worsening the problem of visual-semantic feature inconsistency.

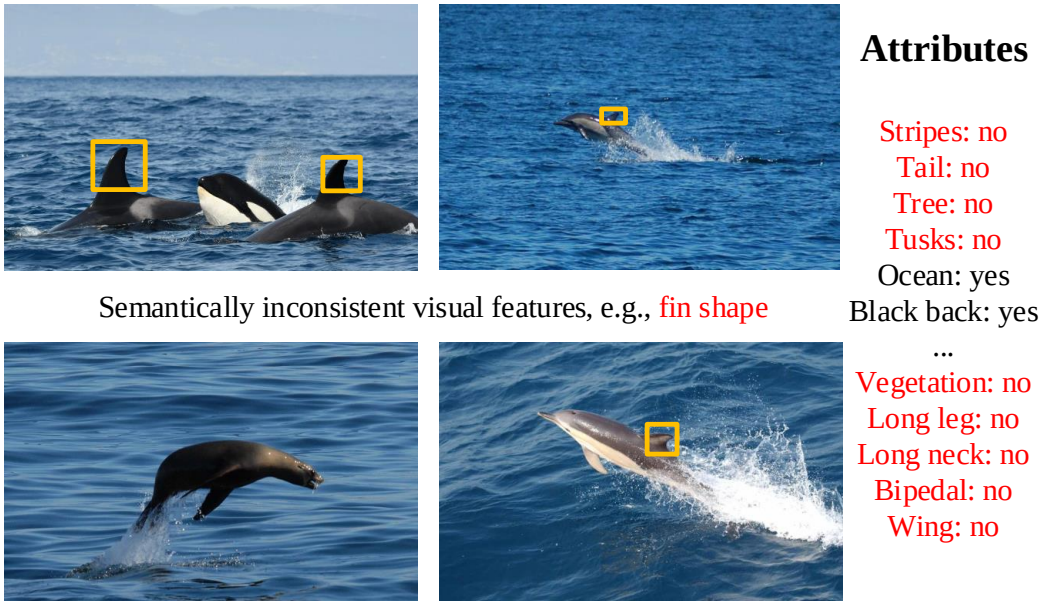

Figure 1. Illustration of visual features inconsistent with annotated attributes (highlighted in yellow boxes) and redundant annotated attributes inconsistent with visual features (highlighted in red text).

We think that extracting and synthesizing refined visual features to enhance the semantic consistency of visual features, and segregating semantic-consistent visual features and visually consistent non-redundant semantic features from raw visual-semantic features to bolster the consistency of visual-semantic features, can alleviate the aforementioned issues. Hence, this paper proposes a GZSL framework that enhances the consistency of visual-semantic features using self-distillation and disentanglement network. Specifically, We first devised a self-distillation module that leverages self-distillation technology to augment both the feature extraction model and the generative model in the context of generative GZSL. This enables them to concurrently acquire refined mid-layer features and soft label knowledge from the auxiliary self-teacher network, thereby stimulating the model to extract and synthesize refined visual features. Additionally, we devise a disentanglement network applied to the visual-semantic modality. For instance, in the visual modality, the visual disentanglement encoder projects visual features into z_r and z_u. To ensure the consistency of z_r with semantic features, visual-semantic features are cross-reconstructed, and a semantic relationship matching method is employed to calculate the compatibility score between z_r and semantic information to guide the learning of z_r.

 doi:10.20944/preprints202404.0672.v1

Furthermore, a latent representation independent method is applied to enforce the independence between z_r and z_u. Ultimately, the disentanglement network attains consistent visual-semantic features, which are amalgamated to jointly train a GZSL classifier.

In summary, the contributions of this paper are as follows:

- We identified that most models typically do not handle visually-semantic inconsistent features and directly align them, which may lead to alignment bias. We propose an approach to enhance the consistency of visually-semantic features by refining visual features and disentangling original visually-semantic features.
- We designed a self-distillation embedding module, which generates soft labels through an auxiliary self-teacher network and employs soft label distillation and feature map distillation methods to refine original visual features of seen classes and synthesized visual features of unseen classes from the generator, thereby enhancing the semantic consistency of visual features.
- We proposed a disentanglement network, which encodes visually-semantic features into latent representations and promotes visually-semantic consistent features to be separated from original features through semantic relation matching and latent representation independence methods, significantly enhancing the consistency of visually-semantic features.
- Extensive experiments on four GZSL benchmark datasets demonstrate that our model can separate refined visually-semantic features with consistency from original visually-semantic features, thereby alleviating alignment bias caused by visually-semantic inconsistency and improving the performance of GZSL models.

2. Related Work

2.1. Generative-Based Generalized Zero-Shot Learning

In recent years, numerous studies have employed generative models to bolster the efficacy of GZSL tasks. GANs or VAEs are commonly utilized in generative GZSL to synthesize visual features for unseen classes. These synthesized visual features for unseen classes are subsequently integrated with original visual features for seen classes to train classifiers. For example, Narayan et al. [32] employed VAEs and GANs to refine the quality of synthesized visual features for unseen classes. They introduced a feedback module to regulate the generator's output, effectively diminishing ambiguity between classes. Zhang et al. [3] combined generative and embedding-based models by projecting real and synthesized samples onto an embedding space for classification, establishing a hybrid ZSL framework that effectively addresses data imbalance issues. Li et al. [2] proposed an innovative approach that integrates a Transformers model with VAE and GAN, capitalizing on the rich data representation from VAE and the diversity of data generated by GAN to mitigate dataset diversity bias, while utilizing Transformers to enhance semantic consistency. DGCNet [4] introduced a Dual Uncertainty Guided Cycle-Consistent Network, which examines the relationship between visual and semantic features through a cycle-consistent embedding framework and dual uncertainty-aware modules, effectively addressing alignment shift problems and enhancing model discriminability and adaptability. However, these methods often ignore the existence of semantically inconsistent visual features and redundant semantic attributes in the original visual-semantic features, which may affect the correct alignment of visual-semantic features. Instead, by first decoupling visually-semantic consistent features before alignment, we have improved the model's accuracy.

2.2. Knowledge Distillation

Knowledge distillation [18] serves as a model compression technique, aiming to reduce the size and computational complexity of a model by transferring knowledge from a complex neural network (referred to as the teacher network) to a smaller neural network (referred to as the student network) . Initially, the concept of knowledge distillation emerged by encouraging the student network to imitate the output log-likelihood of the teacher network [41]. Subsequ ent research introduced intermediate

 doi:10.20944/preprints202404.0672.v1

layer distillation methods, enabling the student network to acquire knowledge from the convolutional layers of the teacher network with feature map-level locality [5–8], or from the penultimate layer of the teacher network [9–13]. However, these methods necessitate pre-training a complex model as the teacher network, a process consuming substantial time and resources. Some recent studies have proposed self-knowledge distillation [45,46], enhancing the training of the student network by leveraging its own knowledge without requiring an additional teacher network. For instance, Zhang et al. [42] segmented the network into several parts and compressed deep-layer knowledge into shallow layers. DLB [43] utilizes instant soft targets generated in the training process of the previous iteration for distillation, achieving performance improvement without altering the model structure. FRSKD [44] introduces an auxiliary self-teacher network to refine knowledge transfer to the student's classifier network, capable of performing self-knowledge distillation using both soft labels and feature map distillation. This paper adopts the concept proposed by FRSKD to construct a self-distillation embedding module, aiming to refine the original seen visual features and the unseen visual features synthesized by the generator.

3. Materials and Methods

3.1. Problem Definition

In GZSL, the dataset comprises visual features X, semantic attributes C, and labels Y, which can be divided into seen classes S and unseen classes U. Specifically, the visual feature set is defined as $X = \{X_S, X_U\}$, and the corresponding label set is represented as $Y = \{Y_S, Y_U\}$, where Y_S and Y_U are disjoint sets. Semantic attributes are defined as $C = \{C_S, C_U\}$. Visual features x_s^i and x_u^i are defined as the ith visual feature, where $x_s^i \in X_S$ and $x_u^i \in X_U$. The corresponding labels for seen and unseen classes are denoted as y_s^i and y_u^i, while c_s^i and c_u^i represent the ith semantic feature, where $c_s^i \in C_S$ and $c_u^i \in C_U$. Thus, the training dataset is defined as $D_s = \left\{x_s^i, c_s^i, y_s^i\right\}_{i=1}^{N^s}$, and the testing dataset is defined as $D_u = \left\{x_u^i, c_u^i, y_u^i\right\}_{i=1}^{N^u}$. The objective of GZSL is to learn a classifier $F_{GZSL} : X \rightarrow Y_S \cup Y_U$.

3.2. Overall Framework

The SDDN architecture primarily comprises two key modules, as depicted in Figure 2. The first module, known as the self-distillation embedding module, utilizes feature fusion techniques and an auxiliary self-teacher network to transfer refined visual features to the student network. It then employs both soft label distillation and feature map distillation to facilitate the generation of refined features by the generative model, thereby enhancing the consistency of visual-semantic features. The second module, referred to as the disentanglement network, employs semantic relationship matching (SRM) method and latent representation independent (IND) method to guide the visual-semantic disentanglement autoencoder in decoupling semantic-consistent visual features and non-redundant semantic features, further strengthening the consistency of visual-semantic features.

 doi:10.20944/preprints202404.0672.v1

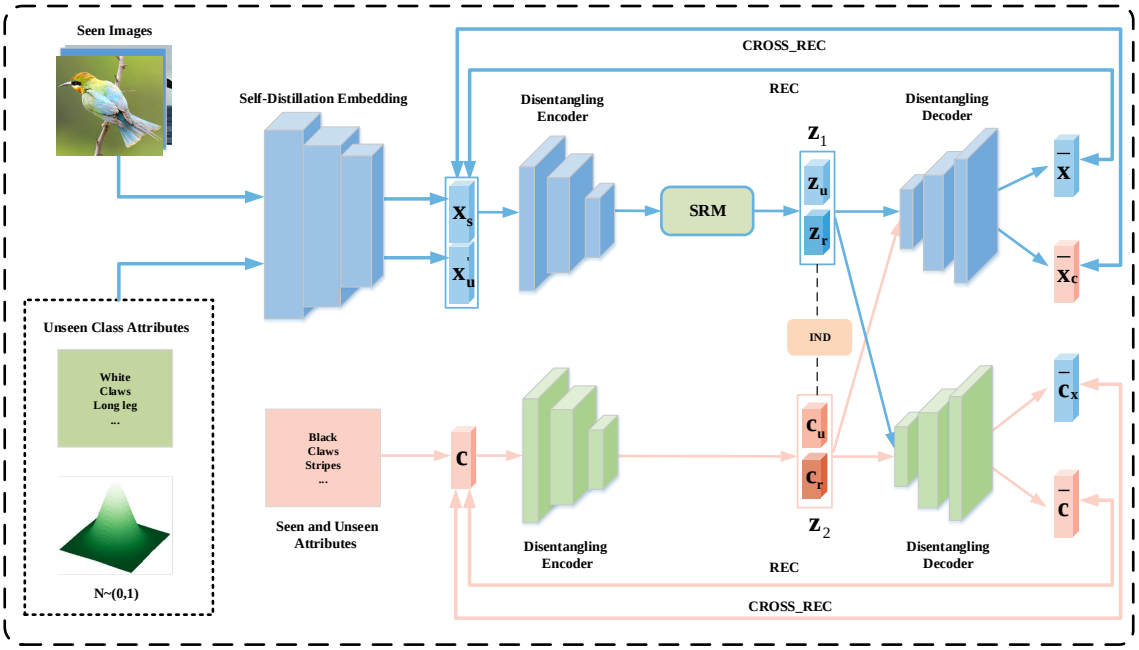

Figure 2. Illustrates of our SDDN framework.

3.3. Self-Distillation Embedding Module

In order to refine the seen visual features extracted by the pre-trained ResNet101 [16] and the unseen visual features synthesized by the generator, thereby improving their semantic consistency. We designed a self-distillation embedding (SDE) module using self-distillation technology, as shown in Figure 3. Firstly, this module establishes an auxiliary self-teacher network, employing specialized feature fusion methods to acquire refined feature maps. Subsequently, it utilizes soft label distillation and feature map distillation to assist the generation model and pre-trained ResNet101 in acquiring knowledge from feature maps and soft labels, refining the extracted and synthesized visual features.

 doi:10.20944/preprints202404.0672.v1

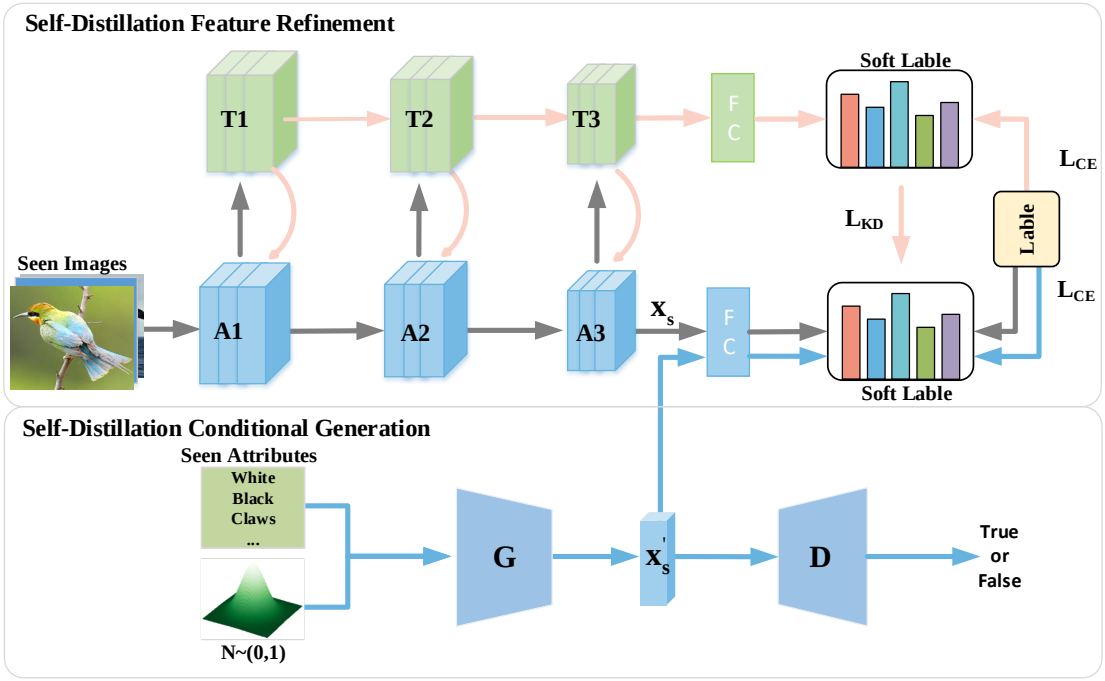

Figure 3. The architecture of self distillation embedded modules. A represents the student network, T represents the auxiliary self-teacher network, G represents the generator, and D represents the discriminator.

3.3.1. Auxiliary Self-Teacher Network

To refine the visual features extracted by the pre-trained ResNet101 [16] and pass them into generation model, we devised an auxiliary self-teacher network T (illustrated in green in Figure 3) based on the architecture proposed by Ji et al. [44], comprising ResNet101, lateral convolutional layers, and feature fusion methods. The auxiliary self-teacher network employs top-down and bottom-up feature fusion to generate refined intermediate feature maps X_{T_i} and produces soft labels P_t at the final layer. Deep neural networks excel at learning representations at various levels; hence, outputs from intermediate and output layers can both contribute to training the student network. In our methodology, we employ ResNet101 (depicted in blue in Figure 3) as the student network A, enriching the visual features extracted by ResNet101 with soft labels P_t from the output of the self-teacher network and refined feature maps X_{T_i} from the intermediate layers. The formula for generating P_t in the auxiliary self-teacher network is defined as follows:

$$P_t = \text{softmax}\left(\frac{\exp(f_t(x^t)/T)}{\sum_{j=1}^{n}\exp(f_t(x_j^t)/T)}\right) \tag{1}$$

Here, T is the temperature parameter (Hinton et al., 2015), typically set to 1. Higher values of T result in softer class probability distributions. The student network learns from P_t through KL divergence, expressed as:

$$\mathcal{L}_{KD}(x^a, P_t, T) = D_{KL}\left(\text{softmax}\left(\frac{\exp(f_a(x^a)/T)}{\sum_{j=1}^{n}\exp(f_a(x_j^a)/T)}\right)||P_t\right) \tag{2}$$

Here, f_a denotes the classifier of the student network. The student network learns refined intermediate layer features $T_i(x^t)$ through the loss function L_F, defined as:

 doi:10.20944/preprints202404.0672.v1

$$\mathcal{L}_F(x^t, x^a) = \sum_{i=1}^{n} ||\phi(T_i(x^t)) - \phi(A_i(x^a))||_2 \tag{3}$$

Where ϕ represents the channel pooling function. Additionally, class predictions are made on the enhanced visual features, and the cross-entropy loss between the prediction results and the true labels is minimized to ensure the accuracy of the enhanced visual features. Finally, the process of refining visual features through the self-learning network can be formalized as:

$$\mathcal{L}_{STN} = \mathcal{L}_{CE}(x^t, y) + \mathcal{L}_{CE}(x^a, y) + \alpha \mathcal{L}_{KD}(x^a, P_t, T) + \beta \mathcal{L}_F(x^t, x^a) \tag{4}$$

3.3.2. Self-Distillation Conditional Generation Module

The Self-Distillation Conditional Generation (SDCG) module aims to train the generator using soft labels and refined visual features generated by the auxiliary self-teacher network, thereby enabling the generator to synthesize refined visual features of unseen classes. The SDCG module employs a conditional generative adversarial network as the generator, utilizing Gaussian noise with a mean of 0 and standard deviation of 1 ($N(0,1)$), and semantic descriptors of seen classes c_s as conditions to synthesize training features ($x_s^{'} = G(c_s, N)$). Next, the synthesized $x_s^{'}$ is input to the trained student network classifier to derive $P_a = f_a(x_s^{'})$. The objective is to minimize the loss between P_{x_s} and the soft labels P_t, ensuring that $x_s^{'}$ remains consistent with the real visual features extracted by the auxiliary self-teacher network. The loss function is as follows:

$$\mathcal{L}_{KD}(x_s^{'}, P_t, T) = D_{KL}\left(softmax\left(\frac{\exp(f_a(x_s^{'})/T)}{\sum_{j=1}^{n}\exp(f_a(x_{s_j}^{'})/T)}\right)||P_t\right) \tag{5}$$

Simultaneously, to ensure the accuracy of the synthesized visual features, the cross-entropy loss between $f_a(x_s^{'})$ and the real labels Y_S is computed. Additionally, the discriminator D is employed to distinguish between real seen class samples (x_s, c_s) and synthesized seen class samples $(x_s^{'}, c_s)$.

$$\mathcal{L}_{wgan}(x_s, x_s^{'}, c_s) = \mathbb{E}[D(x_s, c_s)] - \mathbb{E}[D(x_s^{'}, c_s)] - \lambda \mathbb{E}[(||\nabla_{\hat{x}_s} D(\hat{x}_s, c_s)||_2 - 1)^2] \tag{6}$$

Here, $\hat{x}_s = \alpha x_s + (1-\alpha)x_s^{'}, \alpha \sim U(0,1)$, and λ represents the penalty coefficient. The loss function for the SDCG module is as follows:

$$\mathcal{L}_{SDCG} = \mathcal{L}_{KD}(x_s^{'}, P_t, T) + \mathcal{L}_{wgan}(x_s, x_s^{'}, c_s) + \lambda \mathcal{L}_{CE}(x_s^{'}, Y_S) \tag{7}$$

Finally, the overall loss of the SDE module is $\mathcal{L}_{SDE} = L_{STN} + L_{SDCG}$.

3.4. Disentanglement Network

To further bolster the consistency of visual-semantic features, we propose a disentanglement network. This network utilizes a semantic relation matching method and an independent latent representation method to guide the visual-semantic disentangled autoencoder in separating visually consistent features and non-redundant semantic features from the original data. Furthermore, it aligns these features using a cross-reconstruction method to further strengthen their consistency.

3.4.1. Visual-Semantic Disentangled Autoencoder

The Visual-Semantic Disentangled Autoencoder (VSDA) comprises two parallel variational autoencoders dedicated to processing visual and semantic modalities separately. Each variational autoencoder includes a disentangled encoder and decoder. The disentangled encoder maps the feature space to the latent space, while the decoder maps the latent space back to the feature space. By optimizing the KL divergence loss between the latent variable distribution and the predefined prior

 doi:10.20944/preprints202404.0672.v1

distribution, the VSDA can acquire an effective latent space representation. The KL divergence loss for the VSDA is expressed as:

$$\begin{aligned}\mathcal{L}_{D-VAE} =& \mathbb{E}_{q_\phi(z_1|x)}[\log p(x|z_1)] - D_{KL}(q_\phi(z_1|x)\|p_\theta(z_1)) \\ &+ \mathbb{E}_{q_\phi(z_2|c)}[\log p(c|z_2)] - D_{KL}(q_\phi(z_2|c)\|p_\theta(z_2))\end{aligned} \tag{8}$$

Define visual features $x = \{x_s, x'_u\}$, where x_s represents seen visual features, and x'_u denotes synthetic unseen visual features. Visual feature x and semantic feature c are encoded into latent representations $z_1 = [z_r, z_u]$ and $z_2 = [c_r, c_u]$. Here, z_r and c_r represent the dimensions of semantically consistent visual features and non-redundant semantic features, respectively, while z_u and c_u denote the opposite. This process can be expressed as:

$$\begin{aligned}\boldsymbol{E}_V(x) &= z_1 = [z_r, z_u] \\ \boldsymbol{E}_S(c) &= z_2 = [c_r, c_u]\end{aligned} \tag{9}$$

To mitigate information loss in visual and semantic modalities, a disentangled decoder is employed to reconstruct the latent representations z_1 and z_2 back to the original data and minimize their losses. z_1 is reconstructed into visual features $\tilde{x}$ through the visual unwrapping decoder D_V, and z_2 is reconstructed into semantic features $\tilde{c}$. Their reconstruction loss is computed as follows:

$$\mathcal{L}_{REC} = \sum_{x \in X_S} \sum_{c \in C_S} \|x - \tilde{x}\|^2 + \|c - \tilde{c}\|^2 \tag{10}$$

Simultaneously, to enable the model to learn the association between visual and semantic features and reduce the deviation between modalities, cross-modal cross-reconstruction is designed. This involves using the semantic disentanglement decoder D_S to reconstruct the visual latent representation z_1 and using the visual disentanglement decoder D_V to reconstruct the semantic latent representation z_2. This process can be expressed as:

$$\mathcal{L}_{CROSS\_REC} = \sum_{x \in X_S} \sum_{c \in C_S} |x - D_V(z_2)| + |c - D_S(z_1)| \tag{11}$$

Here, the mean square error (MSE) is utilized to calculate the reconstruction loss between the original visual features and the reconstructed features. Finally, the overall loss of the VSDA is:

$$\mathcal{L}_{VSDA} = \mathcal{L}_{D-VAE} + \mathcal{L}_{REC} + \mathcal{L}_{CROSS\_REC} \tag{12}$$

3.4.2. Semantic Relation Matching

To disentangle the semantically consistent latent representation z_r and the semantically inconsistent latent representation z_u from the original visual space, we adopt the Relation Network (RN) [22] to maximize the Compatibility Score (CS) between the latent representation z_r and the corresponding semantic embedding c to make z_r become semantically consistent. The model structure is shown in Figure 4. The reason why the Relation Network is chosen as the semantic relation matching method here is that the Relation Network can calculate the distance between two samples by building a neural network, thereby measuring the degree of matching between them. The advantage of RN is that it can learn feature embeddings as well as nonlinear metric functions, which capture the similarities between features better than other metric methods. We use RN to concatenate z_r and its unique corresponding semantic embedding c and input it into the relation network R. The score for successful matching is 1, and the score for failed matching is 0. The formula of this process can be expressed as:

$$RS\left(z_{r(t)}, a_{(b)}\right) = \begin{cases} 0, & if y_{(t)} \neq y_{(b)} \\ 1, & if y_{(t)} = y_{(b)} \end{cases} \tag{13}$$

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 10 April 2024 doi:10.20944/preprints202404.0672.v1

Here, t and b respectively represent the tth semantically consistent representation and the bth unique semantic embedding in the training batch, $y_{(t)}$ and $y_{(b)}$ represent the class labels of $c_{(t)}$ and $c_{(b)}$ respectively.

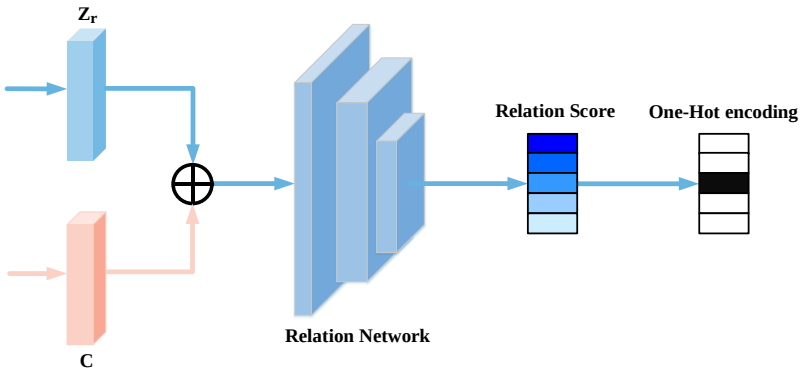

Figure 4. Architecture of semantic relational matching model.

The relation network uses a Sigmoid activation function to constrain the relation score of each pair of outputs between 0 and 1. The loss function for optimizing z_r can be expressed as:

$$\mathcal{L}_{SRM} = \sum_{t=1}^{B}\sum_{b=1}^{n}\left\| RN\left(z_{r(t)}, a_{(b)}\right) - RS\left(z_{r(t)}, a_{(b)}\right)\right\|^2 \tag{14}$$

Here, B represents the size of the training batch, and n represents the number of corresponding unique semantic embeddings in the training batch. In each training batch, calculate the mean square error between the output of the relationship score of each pair $z_{r(t)}$ and $c_{(b)}$ and the ground truth, optimized by the mean square error.This loss ensures that the model disentangle a semantically consistent latent representation z_r.

3.4.3. Independence Between Latent Representations

During the encoding process, we anticipate the latent representation z_1 in the visual modality to encompass visually consistent latent representations z_r with semantics and semantically unrelated latent representations z_u. Similarly, we expect the latent representation z_2 in the semantic modality to include visually consistent latent representations c_r and visually unrelated latent representations c_u. This is crucial for segregating visually-semantic consistent features from the original features. Therefore, we devised a latent representation independence method to foster the segregation of visually-semantic consistent features and visually-semantic unrelated features in the visual-semantic modality. From a probabilistic perspective, z_r and z_u can be regarded as originating from different conditional distributions in the visual modality, while c_r and c_u can be considered to come from different conditional distributions in the semantic modality:

$$\begin{aligned} z_r &\sim \psi_1(z_r|x), \quad z_u \sim \psi_2(z_u|x) \\ c_r &\sim \psi_3(c_r|x), \quad c_u \sim \psi_4(c_u|x) \end{aligned} \tag{15}$$

where ψ_1 and ψ_2 are distributions for z_r and z_u respectively, and ψ_3 and ψ_4 are distributions for c_r and c_u respectively. Thus, their overall independence IND can be expressed as:

$$\begin{aligned} IND_v &= KL(\psi\|\psi_1\cdot\psi_2) = \mathbb{E}_\psi(\log\tfrac{\psi}{\psi_1\cdot\psi_2}) \\ IND_s &= KL(\psi\|\psi_3\cdot\psi_4) = \mathbb{E}_\psi(\log\tfrac{\psi}{\psi_3\cdot\psi_4}) \\ IND &= IND_v + IND_s \end{aligned} \tag{16}$$

 doi:10.20944/preprints202404.0672.v1

where $\psi := \psi(z_r, z_u|x)$ is the joint conditional probability of z_r and z_u, and similarly for the semantic modality. Taking the visual modality as an example, suppose that when $y = 1$, z_r and z_u are dependent, denoted as $\tau(z_1|y = 1)$. While when $y = 0$, z_r and z_u are independent, denoted as $\tau(z_1|y = 0)$. Therefore, IND_v can be represented as:

$$IND_v = \mathbb{E}_\psi(\log \frac{\tau(z_1|y=1)}{\tau(z_1|y=0)}) = \mathbb{E}_\psi(\log \frac{\tau(y=1|z_1)\tau(z_1)/\tau(y=1)}{\tau(y=0|z_1)\tau(z_1)/\tau(y=0)}) = \mathbb{E}_\psi \frac{\tau(y=1|z_1)}{1-\tau(y=1|z_1)}) \tag{17}$$

We introduce a discriminator DIS_v to approximate $\tau(y = 1|z_1)$, thus IND_v can be approximated by the following formula:

$$IND_v = \mathbb{E}_\phi(\log \frac{DIS_v(z_1)}{1 - DIS_v(z_1)}) \tag{18}$$

During the training of the discriminator, we randomly shuffle z_r and z_u in each training batch, then concatenate them to obtain $\hat{z_1}$. Finally, the loss of the discriminator on the visual modality and the semantic modality is given by:

$$\begin{aligned} \mathcal{L}_{DIS} = DIS_v + DIS_s = \log DIS_v(z_1) + \log(1 - DIS_v(\hat{z_1})) \\ + \log DIS_s(z_2) + \log(1 - DIS_s(\hat{z_2})) \end{aligned} \tag{19}$$

In summary, the total loss of our SDDN framework is formulated as:

$$\mathcal{L}_{total} = \mathcal{L}_{SDE} + \mathcal{L}_{VSDA} + \lambda_1 * \mathcal{L}_{SRM} + \lambda_2 * IND + \lambda_3 * \mathcal{L}_{DIS} \tag{20}$$

3.5. Classification

When training the classifier, the seen visual features x_s, refined by the auxiliary self-teacher network, and the synthesized unseen visual features $x_u^{'}$ from the SDCG module are combined into $x = \{x_s, x_u^{'}\}$. Next, x is processed by the trained visual disentangling encoder to yield both semantically consistent latent representation z_r and semantically irrelevant latent representation z_u. Similarly, the semantic attribute set c for both seen and unseen classes is fed into the trained semantic disentangling encoder, producing visually consistent latent representation c_r and visually irrelevant latent representation z_u. Subsequently, these representations z_r and z_u are fused and input into a softmax-based GZSL classifier for category prediction. Finally, the training concludes by minimizing the loss between the real and predicted labels. The whole process is shown in Figure 5.

 doi:10.20944/preprints202404.0672.v1

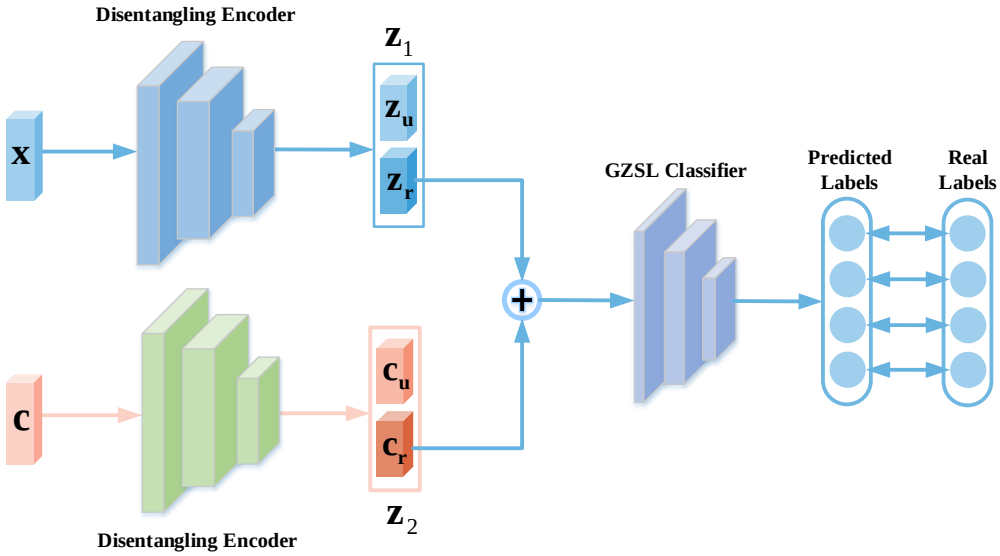

Figure 5. Scheme of classification.

4. Experiments

4.1. Datasets

We conducted comprehensive tests on four publicly available benchmark datasets: Caltech-UCSD Birds-200-2011 (CUB) [39], Animals with Attributes2 (AWA2) [37], SUN Attribute Dataset (SUN) [38], and Oxford Flowers (FLO) [40]. All datasets and their statistics are summarized in Table 1. CUB is a fine-grained dataset comprising 11,788 images from 200 different bird species, with 150 seen classes and 50 unseen classes. Each image in CUB is annotated with 312 dimensions of attributes. FLO is another fine-grained dataset consisting of flower images, containing 8,189 images across 102 classes, including 82 seen classes and 20 unseen classes. The annotation attributes in FLO have 1,024 dimensions. SUN is a fine-grained image dataset featuring various scenes, with 14,340 images covering 717 classes (645 seen classes and 72 unseen classes). Each scene in SUN is associated with 102-dimensional attributes describing its characteristics, such as lighting conditions, weather conditions, and terrain. AWA2 is a coarse-grained dataset with 37,322 animal images across 50 classes (10 seen classes and 40 unseen classes), covering a wide range of animals, including mammals, birds, and reptiles. Each image in AWA2 is labeled with attributes of dimension 85.

Table 1. Statistics of the AWA2,CUB,FLO and SUN datasets, including visual feature dimension D^x, semantic feature dimension D^s, num-ber of seen classes N^s, number of unseen classes N^u and number of all instances N^i.

Dataset	D^x	D^s	N^s	N^u	N^i
AWA2	2048	85	40	10	37322
CUB	2048	312	150	50	11788
FLO	2048	1024	82	20	8189
SUN	2048	102	645	72	14340

4.2. Evaluation Protocol

During testing, the accuracy is assessed on the test sets for both seen classes (S) and unseen classes (U). Here, U represents the average accuracy for each class on test images of unseen classes, indicating the model's ability to classify samples from previously unseen classes. S represents the average accuracy for each class on test images of seen classes, reflecting the model's ability to classify

 doi:10.20944/preprints202404.0672.v1

samples from seen classes. H (defined as $(H = (2 \times S \times U)/(S + U))$ represents the harmonic mean of S and U, serving as an evaluation metric for the performance of GZSL classification.

4.3. Implementation Details

SDDN mainly consists of a SDE module and a disentanglement network. The SDE module mainly consists of a student network, a self-teacher network, a generator and a discriminator. The student network is a ResNet101 model pre-trained on ImageNet and is used to extract visual features with a dimension of 2048. The self-teacher network is composed of the student network itself and the feature fusion method. In addition, the generator is implemented using a multi-layer perceptron with a hidden layer dimension of 2048, and the discriminator is implemented using a fully connected layer and activation function. The disentanglement network consists of an encoder, a decoder, a discriminator and a semantic relationship matching model. Both the encoder and decoder are multi-layer perceptrons with a single hidden layer and 2048 hidden units. The semantic relationship matching model consists of two fully connected layers activated with Smooth Maximum Unit (SMU) [47] activation function and Sigmiod function respectively. The discriminator is implemented using a fully connected layer and SMU activation function.

The hardware environment used by SDDN is Intel i7-10700K CPU, RTX A5000 32GB GPU; the software environment is Ubuntu 20.04 LTS operating system, cuda 11.4.0, and cudnn 8.2.4. SDDN is implemented in PyTorch 1.10.1. The ADAM optimizer is used to optimize the parameters of each module. The learning rate of the Adam optimizer is set to $lr = 0.0001$, $\beta_1 = 0.9$ and $\beta_2 = 0.999$. The batch size is 64. The loss weight λ_1 of the semantic relation matching method, the loss weight λ_2 of the latent representation independence method and the weight λ_3 of the visual-semantic discriminator are set between 0.1-25.

4.3.1. Comparing with the State-of-the-Arts

To validate the effectiveness of our proposed SDDN model, we computed the seen class accuracy rate S, unseen class accuracy rate U, and their harmonic mean H on the aforementioned four datasets. We compared them with 15 state-of-the-art models, and the comparison results are shown in Table 2. These 15 models are categorized into methods based on generative models and methods not based on generative models. Generative-based methods typically utilize techniques such as GANs or VAEs to generate synthetic unseen class data to augment the training dataset. These synthetic data can be used to train models in ZSL to improve their generalization capability to unseen classes. Non-generative-based methods, on the other hand, do not rely on generating synthetic data but achieve generalization to unseen classes through techniques such as feature embedding and alignment of existing data. Our method belongs to the generative-based methods.

 doi:10.20944/preprints202404.0672.v1

Table 2. Performance comparison in accuracy(%) on four datasets. Displaying the accuracies of seen and unseen classes in GZSL, denoted as U, S, and H for the harmonic mean. The methods above and below the horizontal line correspond to non-generative and generative approaches, respectively. Results in **bold** font indicate the highest performance.

Method	FLO			CUB			AWA2			SUN		
	U	*S*	*H*	*U*	*S*	*H*	*U*	*S*	*H*	*U*	*S*	*H*
TCN [15]	-	-	-	52.6	52.0	52.3	61.2	65.8	63.4	31.2	37.3	34.0
DVBE [23]	-	-	-	53.2	60.2	56.5	63.6	70.8	67.0	45.0	37.2	40.7
RGEN [24]	-	-	-	60.0	**73.5**	66.1	67.1	76.5	71.5	44.0	31.7	36.8
TDCSS [34]	54.1	85.1	66.2	44.2	62.8	51.9	59.2	74.9	66.1	-	-	-
f-VAEGAN-D2 [28]	56.8	74.9	64.6	48.4	60.1	53.6	57.6	70.6	63.5	45.1	38.0	41.3
LisGAN [30]	57.7	83.8	68.3	46.5	57.9	51.6	52.6	76.3	62.3	42.9	37.8	40.2
CANZSL [29]	58.2	77.6	66.5	47.9	58.1	52.5	49.7	70.2	58.2	46.8	35.0	40.0
SE-GZSL [31]	-	-	-	41.5	53.3	46.7	58.3	68.1	62.8	30.5	40.9	34.9
TIZSL [33]	70.4	68.7	69.5	52.1	53.3	52.7	**76.8**	66.9	71.5	32.3	24.6	27.9
FREE [19]	67.4	84.5	75.0	55.7	59.9	57.7	60.4	75.4	67.1	47.4	37.2	41.7
SDGZSL [17]	83.3	90.2	86.6	59.9	66.4	63.0	64.6	73.6	68.8	48.2	36.1	41.3
JG-ZSL [3]	-	-	-	60.8	63.9	62.3	63.1	68.3	65.6	**50.2**	37.9	43.2
ICCE [36]	66.1	86.5	74.9	**67.3**	65.5	66.4	65.3	**82.3**	72.8	-	-	-
DGCNet-db [4]	-	-	-	51.5	57.5	54.4	50.4	72.8	59.6	26.8	39.6	32.0
DVAGAN [2]	-	-	-	52.5	57.3	54.8	65.9	82.0	**73.1**	44.7	37.9	41.0
Our SDDN	**87.3**	**90.5**	**88.9**	66.8	68.3	**67.5**	65.6	74.3	69.7	48.6	**42.3**	**45.2**

From the comparison results in the table, firstly, our SDDN achieved the highest accuracy on U, S, and H on the FLO dataset, surpassing all compared models. Specifically, we outperformed the second-best model by 2.3% in the H metric. There was a significant improvement in the U metric, where we led the second-best by 4%. In the S metric, we were ahead of the second-best by 0.3%. On the CUB dataset, we achieved the highest accuracy on the U metric, leading the second-best by 1.1%. Additionally, we obtained the second-best accuracy on both U and S metrics, leading the third-best by 6% and 1.9%, respectively. On the SUN dataset, we attained the highest accuracy on both S and H metrics, leading the second-best by 2% in the H metric and 1.4% in the S metric.

Overall, our performance was the best in the H metric on these three fine-grained datasets: FLO, CUB, and SUN, the best on U in FLO, and the best on S in both FLO and SUN. This indicates that the richer the information in the dataset, the more effectively our proposed method can capture it through self-distillation and disentanglement techniques, separating visually-semantic consistent features and aligning them effectivel.

5. Model Analysis

5.1. Ablation Study

In our ablation study, we aim to isolate the key components of SDDN and assess their impact on GZSL. We remove the Semantic relation matching loss (LSRM) to evaluate the contribution of the Visual-Semantic Matching module to extracting semantically consistent visual features. Omitting the Independence score (IND) allows us to evaluate its contribution to further separating visually-semantic consistent features. Additionally, we exclude the loss of the self-distillation embedding module (LSDE) and then used the pre-trained ResNet101 and regular generator without employing the self-distillation technique. This evaluation helps assess the refinement effect of SDE on extracting and synthesizing visual features, while validating the effectiveness of the disentanglement network. Our ablation experiments were conducted on the FLO and CUB datasets, with the experimental results presented in Table 3 and Figure 6.

 doi:10.20944/preprints202404.0672.v1

Table 3. Ablation study of different component combinations on FLO and CUB datasets. Results are reported in %, with the best results highlighted in bold.

Method	FLO			CUB		
	U	S	H	U	S	H
SDDN w/o LSRM	60.3	71.6	65.5	48.1	57.8	52.5
SDDN w/o IND	79.6	81.2	80.4	56.7	60.9	58.7
SDDN w/o LSDE	86.9	88.7	87.8	65.5	67.9	66.7
SDDN	**87.3**	**90.5**	**88.9**	**66.8**	**68.2**	**67.5**

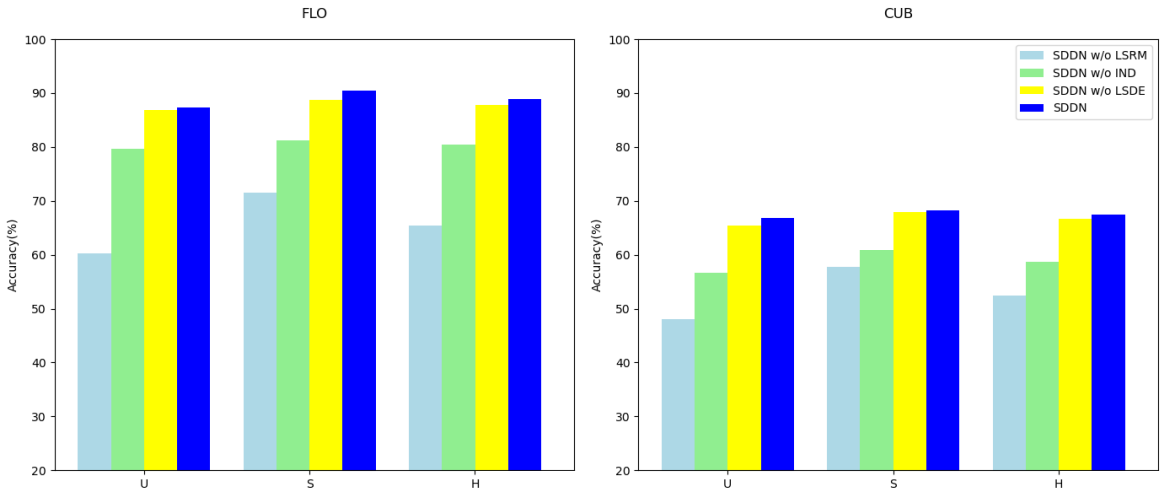

Figure 6. Ablation study on different components combinations of the FLO and CUB datasets.

The results underscore the critical importance of the semantic relation matching module (LSRM), Independence score (IND), and self-distillation embedding module (LSDE) for the performance of SDDN. Firstly, LSRM is particularly crucial for visual-semantic feature alignment, as its removal leads to a significant decrease in the accuracy of seen classes (S), unseen classes (U), and the harmonic mean (H). Secondly, IND is essential for further separating visually-semantic consistent features from the original features, as its removal results in lower U, S, and H values. Additionally, LSDE helps refine the original visual features of seen classes and the synthesized features of unseen classes, as the model without LSDE performs lower in U, S, and H compared to the complete SDDN. Furthermore, when comparing Table 1, it is found that the model without SDE still achieves the highest H score on FLO and CUB, indicating the effectiveness of the disentanglement network. Finally, the complete SDDN model demonstrates superior performance across all metrics, proving its effectiveness in GZSL.

5.2. Hyper-Parameter Analysis

In this study, the optimization objective of SDDN is determined by three critical hyperparameters: the coefficient of semantic relationship matching loss (λ_1), the coefficient of independence of latent representations (λ_2), and the coefficient of discriminator loss (λ_3). To elucidate the influence of each hyperparameter on model performance, sensitivity analysis was conducted by varying the hyperparameter values in the experiments. Specifically, λ_1 was varied within the range of 0.3-20.0, while λ_2 and λ_3 were varied within the range of 0.1-3.0. Figure 7 illustrates the significant impact of hyperparameter values λ_1, λ_2, and λ_3 on the experimental outcomes. Notably, when λ_1 is set to 18, λ_2 is set to 0.5, and λ_3 is set to 2, the model achieves its highest accuracy on the FLO dataset. Whereas, when λ_1 is set to 1, λ_2 is set to 0.6, and λ_3 is set to 0.3, the model achieves its highest accuracy on the CUB dataset. These observations underscore the substantial influence of hyperparameter

 doi:10.20944/preprints202404.0672.v1

weights on model accuracy, indicating the model's high sensitivity to these hyperparameters. Based on these findings, we advocate for future experiments to focus on exploring the specific impact of minor fluctuations in these three hyperparameter values on accuracy. This systematic analysis of hyperparameters will contribute to a deeper comprehension of model behavior and offer valuable insights for optimizing model performance across diverse datasets.

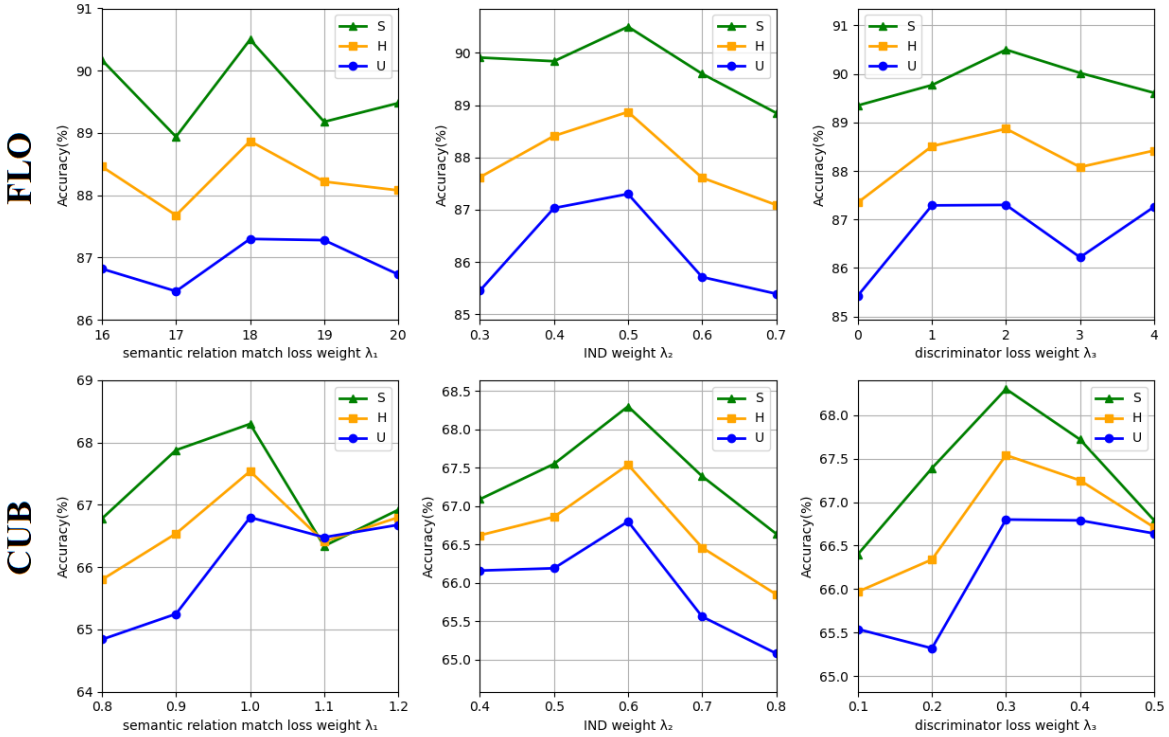

Figure 7. Hyperparameter Analysis: The impact of values for weights λ_1, λ_2 and λ_3 on model performance is examined.

5.3. Zero-Shot Retrieval Performance

To assess the practical application performance of our SDDN framework, we conducted zero-shot retrieval experiments comparing SDDN with two other state-of-the-art generative-based GZSL frameworks: DGGNet-db and DVAGAN. The experiment follows the zero-shot retrieval protocol in SDGZSL [17]. In zero-shot retrieval experiments, we initially provide semantic features of unseen classes, followed by employing the generation modules of SDDN, DGGNet-db, and DVAGAN to synthesize a certain number of visual features for these unseen classes. Throughout this process, the average of the synthesized visual features for each category is computed as the retrieval feature. Subsequently, the cosine similarity between the retrieval features and the true features is calculated, and the true features are ranked in descending order based on this similarity. The performance of zero-shot image retrieval is evaluated using mean Average Precision (mAP). Experimental analyses are performed on three datasets: CUB, AWA2, and SUN. The results, illustrated in Figure 8, compare SDDN, DGGNet-db, and DVAGAN in terms of zero-shot retrieval performance. The horizontal coordinates 100, 50, and 25 represent the proportions of unseen category images in the test dataset, being 100%, 50%, and 25%, respectively, while the vertical coordinate represents the average retrieval accuracy. Results indicate significantly higher zero-shot retrieval performance of the SDDN framework on the CUB and SUN datasets compared to DGGNet-db and DVAGAN. In the AWA2 dataset, we achieve the best performance when the proportion of unseen category images reaches 50%, and it remains close to the best performance when the proportion reaches 100%. These zero-shot retrieval performance tests on the three datasets further validate the effectiveness of the model.

 doi:10.20944/preprints202404.0672.v1

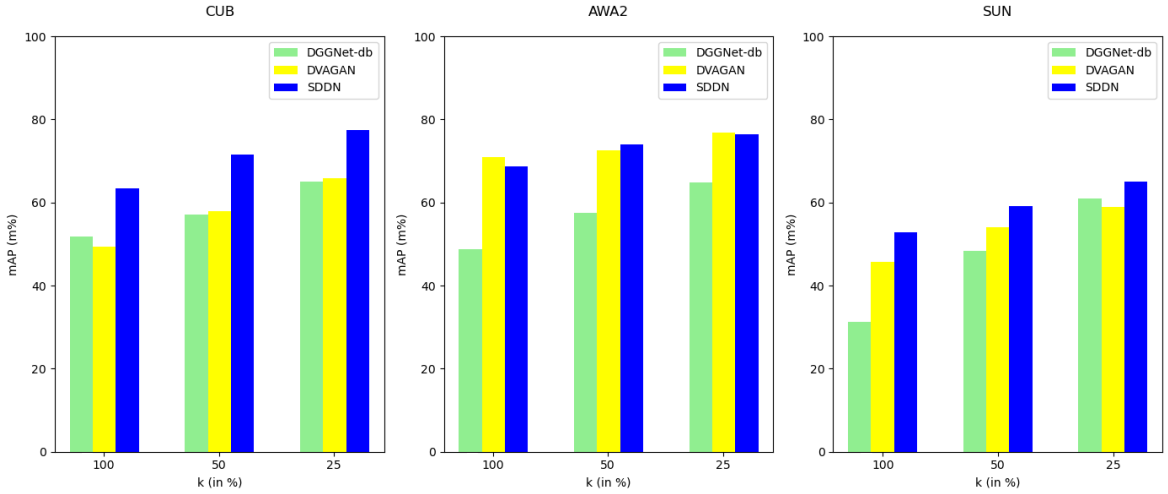

Figure 8. Comparison of Zero-shot Retrieval Performance.

6. Conclusion

In this paper, we propose a generalized zero-shot learning framework that utilizes self-distillation and disentanglement network to enhance visual-semantic feature consistency. Initially, for improving the semantic consistency of visual features, we develop a self-distillation embedding framework integrating self-distillation techniques with a conditional generator to prompt the synthesis of refined visual features. Subsequently, to further promote visual-semantic feature consistency, we design a disentanglement network. We use semantic relation matching networks and latent representation independence methods to facilitate the separation of visually semantically consistent features from inconsistent features. Additionally, we devise a cross-reconstruction method to align visual and semantic features within a visual-semantic common space, thereby enhancing the semantic consistency of visual-semantic features. Extensive experiments are conducted on four widely used benchmark datasets in GZSL. We compare SDNN with current state-of-the-art methods, thereby demonstrating the superiority of the proposed SDNN framework. In future work, we intend to optimize the model further and apply it in the field of medical diagnostics to assist in identifying new disease patterns.

Author Contributions: Responsible for proposing research ideas,modeling frameworks, content planning, guidance,and full-text revisions: X.L.;Responsible for literature research, research methodology, experimental design, thesis writing, and full text revision: C.W.;Responsible for lab instruction,guidelines,and full text revisions:G.Y. and CH.W.;Responsible for providing guidance, revising,and reviewing full texts:,Y.L.,J.L. and Z.Z.All authors have read and agreed to the published version ofthe manuscript.

Funding: This work was supported by National Key Research and Development Program of China (2020AAA0109700), National Science and Technology Major Project (2020AAA0109703), National Natural Science Foundation of China (62076167, U23B2029), the Key Scientific Research Project of Higher Education Institutions in Henan Province (24A520058, 24A520060) and Postgraduate Education Reform and Quality Improvement Project of Henan Province (YJS2024AL053).

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare that they have no conflict of interest.

References

1. Z. Wang, Y. Hao, T. Mu, O. Li, S. Wang, and X. He, "Bi-directional distribution alignment for transductive zero-shot learning," 2023.
2. N. Li, J. Chen, N. Fu, W. Xiao, T. Ye, C. Gao, and P. Zhang, "Leveraging dual variational autoencoders and generative adversarial networks for enhanced multimodal interaction in zero-shot learning," *Electronics*, vol. 13, no. 3, p. 539, 2024.
3. M. Zhang, X. Wang, Y. Shi, S. Ren, and W. Wang, "Zero-shot learning with joint generative adversarial networks," *Electronics*, vol. 12, no. 10, p. 2308, 2023.

 doi:10.20944/preprints202404.0672.v1

4. Y. Zhang, Y. Tian, S. Zhang, and Y. Huang, "Dual-uncertainty guided cycle-consistent network for zero-shot learning," *IEEE Transactions on Circuits and Systems for Video Technology*, 2023.
5. S. Zagoruyko and N. Komodakis, "Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer," 2017.
6. J. Kim, S. Park, and N. Kwak, "Paraphrasing complex network: Network compression via factor transfer," 2020.
7. J. Yim, D. Joo, J. Bae, and J. Kim, "A gift from knowledge distillation: Fast optimization, network minimization and transfer learning," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 7130–7138.
8. A. Koratana, D. Kang, P. Bailis, and M. Zaharia, "Lit: Learned intermediate representation training for model compression," in *International Conference on Machine Learning*. PMLR, 2019, pp. 3509–3518.
9. F. Tung and G. Mori, "Similarity-preserving knowledge distillation," 2019.
10. B. Peng, X. Jin, J. Liu, S. Zhou, Y. Wu, Y. Liu, D. Li, and Z. Zhang, "Correlation congruence for knowledge distillation," 2019.
11. Y. Tian, D. Krishnan, and P. Isola, "Contrastive representation distillation," 2022.
12. Y. Liu, J. Cao, B. Li, C. Yuan, W. Hu, Y. Li, and Y. Duan, "Knowledge distillation via instance relationship graph," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.
13. W. Park, D. Kim, Y. Lu, and M. Cho, "Relational knowledge distillation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.
14. K. Sun, X. Zhao, H. Huang, Y. Yan, and H. Zhang, "Boosting generalized zero-shot learning with category-specific filters," *Journal of Intelligent & Fuzzy Systems*, no. Preprint, pp. 1–14.
15. H. Jiang, R. Wang, S. Shan, and X. Chen, "Transferable contrastive network for generalized zero-shot learning," 2019.
16. K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015.
17. Z. Chen, Y. Luo, R. Qiu, S. Wang, Z. Huang, J. Li, and Z. Zhang, "Semantics disentangling for generalized zero-shot learning," 2021.
18. Z. Han, Z. Fu, S. Chen, and J. Yang, "Contrastive embedding for generalized zero-shot learning," 2021.
19. S. Chen, W. Wang, B. Xia, Q. Peng, X. You, F. Zheng, and L. Shao, "Free: Feature refinement for generalized zero-shot learning," 2021.
20. X. Li, Z. Xu, K.-J. Wei, and C. Deng, "Generalized zero-shot learning via disentangled representation," in *AAAI Conference on Artificial Intelligence*, 2021.
21. B. Tong, C. Wang, M. Klinkigt, Y. Kobayashi, and Y. Nonaka, "Hierarchical disentanglement of discriminative latent features for zero-shot learning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 11 467–11 476.
22. F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H. S. Torr, and T. M. Hospedales, "Learning to compare: Relation network for few-shot learning," 2018.
23. S. Min, H. Yao, H. Xie, C. Wang, Z.-J. Zha, and Y. Zhang, "Domain-aware visual bias eliminating for generalized zero-shot learning," 2020.
24. G.-S. Xie, L. Liu, F. Zhu, F. Zhao, Z. Zhang, Y. Yao, J. Qin, and L. Shao, "Region graph embedding network for zero-shot learning," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IV 16*. Springer, 2020, pp. 562–580.
25. Y. Liu, L. Zhou, X. Bai, Y. Huang, L. Gu, J. Zhou, and T. Harada, "Goal-oriented gaze estimation for zero-shot learning," 2021.
26. C. Wang, S. Min, X. Chen, X. Sun, and H. Li, "Dual progressive prototype network for generalized zero-shot learning," 2021.
27. C. Wang, X. Chen, S. Min, X. Sun, and H. Li, "Task-independent knowledge makes for transferable representations for generalized zero-shot learning," 2021.
28. Y. Xian, S. Sharma, B. Schiele, and Z. Akata, "f-vaegan-d2: A feature generating framework for any-shot learning," 2019.
29. Z. Chen, J. Li, Y. Luo, Z. Huang, and Y. Yang, "Canzsl: Cycle-consistent adversarial networks for zero-shot learning from natural language," 2019.

 doi:10.20944/preprints202404.0672.v1

30. J. Li, M. Jin, K. Lu, Z. Ding, L. Zhu, and Z. Huang, "Leveraging the invariant side of generative zero-shot learning," 2019.
31. J. Kim, K. Shim, and B. Shim, "Semantic feature extraction for generalized zero-shot learning," 2021.
32. S. Narayan, A. Gupta, F. S. Khan, C. G. M. Snoek, and L. Shao, "Latent embedding feedback and discriminative features for zero-shot classification," 2020.
33. L. Feng and C. Zhao, "Transfer increment for generalized zero-shot learning," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, pp. 2506–2520, 2020.
34. Y. Feng, X. Huang, P. Yang, J. Yu, and J. Sang, "Non-generative generalized zero-shot learning via task-correlated disentanglement and controllable samples synthesis," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 9346–9355.
35. G. Kwon and G. AlRegib, "A gating model for bias calibration in generalized zero-shot learning," 2022.
36. X. Kong, Z. Gao, X. Li, M. Hong, J. Liu, C. Wang, Y. Xie, and Y. Qu, "En-compactness: Self-distillation embedding & contrastive generation for generalized zero-shot learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 9306–9315.
37. Y. Xian, C. H. Lampert, B. Schiele, and Z. Akata, "Zero-shot learning – a comprehensive evaluation of the good, the bad and the ugly," 2020.
38. G. Patterson and J. Hays, "Sun attribute database: Discovering, annotating, and recognizing scene attributes," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 2751–2758.
39. P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona, "Caltech-ucsd birds 200," 2010.
40. M.-E. Nilsback and A. Zisserman, "Automated flower classification over a large number of classes," in *2008 Sixth Indian conference on computer vision, graphics & image processing*. IEEE, 2008, pp. 722–729.
41. G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, 2015.
42. L. Zhang, J. Song, A. Gao, J. Chen, C. Bao, and K. Ma, "Be your own teacher: Improve the performance of convolutional neural networks via self distillation," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 3713–3722.
43. Y. Shen, L. Xu, Y. Yang, Y. Li, and Y. Guo, "Self-distillation from the last mini-batch for consistency regularization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 11 943–11 952.
44. M. Ji, S. Shin, S. Hwang, G. Park, and I.-C. Moon, "Refine myself by teaching myself: Feature refinement via self-knowledge distillation," 2021.
45. M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin, "Emerging properties in self-supervised vision transformers," 2021.
46. T.-B. Xu and C.-L. Liu, "Data-distortion guided self-distillation for deep neural networks," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, pp. 5565–5572, 07 2019.
47. K. Biswas, S. Kumar, S. Banerjee, and A. K. Pandey, "Smooth maximum unit: Smooth activation function for deep networks using smoothing maximum technique," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 794–803.