Optimized Autoencoder-Driven Semantic Feature Enhancement for Zero-Shot Image Classification

Shaista Khanam*, Poonam Sonar

Rajiv Gandhi Institute of Technology, Maharashtra, India

Received 06 March 2026; revised 19 May 2026; accepted 25 May 2026

DOI: https://doi.org/10.46604/aiti.2026.16231

Abstract

Zero-shot learning (ZSL) identifies unseen categories using semantic knowledge transferred from seen classes. Its effectiveness depends on visual and semantic representations. This study aims to develop an optimized autoencoder-driven semantic feature extraction (OADSFE) framework based on a hybrid feature approach (HFA). The HFA combines deep spatial representations with multi-scale texture information to characterize visual data. Semantic features are derived using fastText, GloVe, BERT, and MPNet, which are evaluated independently. An autoencoder-based post-embedding optimization module compresses high-dimensional semantic embeddings into a compact latent space while preserving discriminative information, reducing memory usage and testing time. Evaluation on the AWA2, SUN, and CUB benchmark datasets demonstrates that the proposed framework achieves up to a 16.29% reduction in testing time and an 89.42% reduction in memory usage while maintaining classification performance across multiple embedding configurations. The proposed framework performs well across diverse datasets and semantic embedding strategies, indicating its suitability for scalable ZSL applications.

Keywords: autoencoder, language models, semantic feature optimization, memory consumption, time complexity

1. Introduction

Conventional image classification techniques achieve strong performance across diverse applications but generally require large amounts of labeled training data. These models are limited to recognizing only the classes observed during training and cannot generalize to unseen categories. Zero-shot learning (ZSL) addresses this limitation by enabling recognition of unseen classes using auxiliary semantic information such as attributes or textual descriptions that establish relationships between seen and unseen categories. In ZSL, semantic representations are associated with visual features to facilitate the prediction of unseen classes through similarity-based inference [1].

Extensive studies have explored this area, focusing on two main ZSL methods: generative and embedding-based. Generative ZSL approaches [2-5] use architectures such as generative adversarial networks (GANs) to generate artificial samples of classes not seen during training. Discriminative features for unseen classes are generated by conditioning on their semantic descriptions. These approaches effectively transform the ZSL task into a conventional supervised classification problem. Embedding-based ZSL, on the other hand, focuses on learning visual and semantic features and aligning them within a shared embedding space [6-9].

Although deep learning has significantly improved visual recognition, existing ZSL models still struggle when unseen classes exhibit subtle visual differences and overlapping semantics [3, 10]. In prior work, the hybrid feature approach (HFA) [11] addressed this challenge by enriching deep spatial (DS) CNN representations with multi-scale texture (MST) descriptors

* Corresponding author. E-mail address: shaista.khan@vcet.edu.in

to better discriminate visually similar classes. While this hybrid visual representation improved discrimination, HFA relied on a single pretrained language model (fastText) to encode semantic information and used the embeddings in their raw high-dimensional form. Thus, despite improvements in visual discrimination, the semantic branch of HFA remains insufficiently optimized and has not been fully exploited.

Recent advances in natural language processing [12] have produced more powerful semantic models, such as GloVe, BERT, and MPNet, that capture richer semantic relationships. Transformer-based architectures recently attracted increasing attention in ZSL for their ability to model long-range semantic dependencies and improve visual–semantic alignment through self-attention mechanisms. Hybrid routing transformer frameworks have been explored to enhance semantic interaction and cross-modal representation learning for ZSL tasks [13]. Subsequently, vision transformer (ViT)-based ZSL approaches demonstrated improved unseen-class recognition by progressively integrating semantic guidance into transformer feature learning [14]. Swin transformer-based shifted-window self-attention models and multi-head self-attention ViT frameworks have also been investigated to improve discriminative feature representation and semantic association in generalized zero-shot learning (GZSL) settings [15-16].

However, these models often produce high-dimensional semantic embeddings that may increase computational cost and introduce redundant semantic information when used directly in zero-shot image classification (ZSIC) frameworks. Although previous studies have primarily concentrated on selecting an appropriate word representation model, the post-embedding dimensionality optimization of semantic features remains largely underexplored in ZSL.

Conventional dimensionality reduction approaches, such as principal component analysis (PCA), project features onto linear subspaces, limiting their ability to capture the nonlinear structure of semantic embeddings. Autoencoders [17] provide a nonlinear framework for learning compact latent semantic representations. Several ZSL methods have incorporated autoencoders for visual–semantic mapping and reconstruction. Kodirov et al. [18] proposed a semantic autoencoder (SAE) for bidirectional visual–semantic projection, while subsequent extensions further enhanced semantic reconstruction and discriminative representation learning through bi-shifting and dual-semantic-constraint strategies [19-20]. Liu et al. [21] introduced a low-rank embedded SAE (LESAE) with low-rank constraints on the semantic space to preserve intrinsic semantic structure and improve generalization, and later rank-controlled semantic autoencoder frameworks further addressed projection-domain shift and unseen-class representations [22].

Heyden et al. [23] further proposed an integral projection-based SAE (IP-SAE) to improve cross-modal reconstruction, demonstrating the continued advancement of semantic autoencoder frameworks for zero-shot learning. However, these approaches primarily focus on visual–semantic mapping and reconstruction rather than dedicated post-embedding semantic optimization. As summarized in Table 1, existing SAE-based methods either use raw semantic embeddings directly or provide limited dimensionality reduction, which may retain redundant semantic information and restrict semantic discriminability for unseen classes.

Table 1 Comparison of semantic feature learning methods

Property	SAE [18]	LESAE [21]	IP-SAE [23]	OADSFE
Post-embedding optimization	×	×	×	✓
Dimensionality reduction	×	✓	×	✓
Semantic refinement	×	×	×	✓

To address this limitation, this study proposes an optimized autoencoder-driven semantic feature extraction (OADSFE) framework as an enhancement of HFA for ZSIC. The proposed approach evaluates multiple semantic models and employs an autoencoder-based semantic compression stage to transform high-dimensional semantic embeddings into compact latent representations while reducing redundancy. Unlike existing SAE-based methods that primarily focus on visual–semantic

mapping and reconstruction, OADSFE explicitly performs post-embedding semantic optimization before classification. The proposed framework is experimentally validated on benchmark datasets, including SUN [24], AWA2 [25], and CUB [26], achieving effective classification performance with reduced memory usage and testing time. The key contributions of this work are summarized as follows:

(1) Semantic feature enhancement and Optimization: The proposed framework enhances semantic representations in HFA by evaluating multiple pre-trained embedding models and introducing an autoencoder-based optimization stage for dimensionality reduction and redundancy removal.
(2) Memory-and Time-Efficient ZSL Framework: By compressing high-dimensional semantic embeddings, the proposed OADSFE reduces memory usage and inference time while preserving discriminative capability.
(3) Scalable and dataset-adaptive design: The framework is compatible with multiple datasets and embedding dimensions, illustrating its adaptability to diverse ZSL scenarios.
(4) Accuracy-efficiency trade-off analysis: OADSFE explicitly analyzes the trade-off between semantic compactness and classification accuracy, demonstrating reductions in memory usage and inference time with only marginal performance variation in some cases.

This paper is organized into four sections. The enhanced HFA approach and proposed OADSFE framework are introduced in Section 2. Experimental findings, analysis, and results are presented in Section 3, followed by conclusions in Section 4.

2. Methodology

The proposed methodology consists of two major components: Enhanced HFA and the OADSFE framework for ZSIC. The proposed Enhanced HFA improves visual–semantic representation learning by integrating deep spatial (DS) and multi-scale texture (MST) features with multiple semantic embedding models, including fastText, GloVe, BERT, and MPNet. Furthermore, the proposed OADSFE framework enhances semantic representations through autoencoder-based dimensionality optimization and redundancy reduction.

2.1 The proposed Enhanced HFA

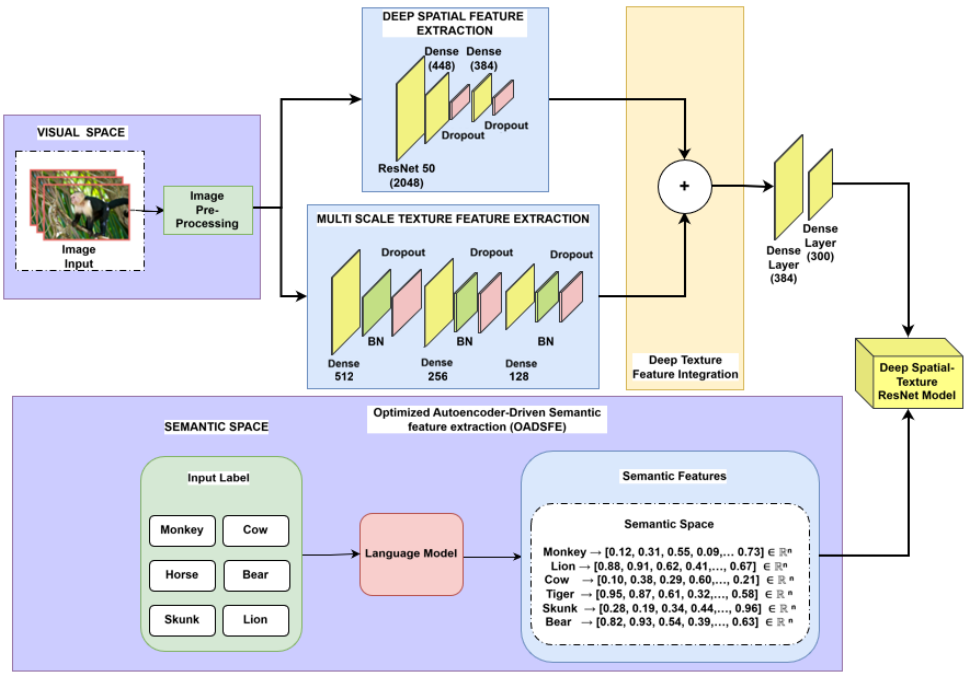


Fig. 1 Proposed Enhanced HFA for ZSIC (Adapted from [4])

The proposed Enhanced HFA extends the earlier HFA approach for ZSIC [11], as illustrated in Fig. 1. The original HFA approach was designed to enhance visual feature representation by leveraging both DS and MST features. Semantic representations in HFA were restricted to fastText embeddings. In the present work, an enhanced version of HFA is proposed, in which the semantic space is enriched using multiple semantic models, including fastText, BERT, GloVe, and MPNet, to enable a comprehensive comparison of the most effective embeddings. Additionally, the proposed OADSFE framework refines semantic representations by reducing redundancy in the embeddings.

Fig. 1 depicts the input image that undergoes pre-processing, including resizing and normalization, which can be expressed as:

$$I' = \frac{I - \mu}{\sigma} \tag{1}$$

where I' represents the normalized image, μ is the mean value of the pixels, and σ is the standard deviation. Next, visual features are extracted using ResNet50 ($f_{encoder}$) to obtain deep spatial features F_s using the equation below:

$$F_s = f_{\text{ResNet50}}(I') \tag{2}$$

The MST features F_T are calculated by:

$$F_T = \sum_k G_{\text{abork}} * I'_k + \sum_m G_{\text{LCM}m} * I'_m \tag{3}$$

where G_{abork} represents the K^{th} Gabor filter applied to the input image I' and G_{LCMm} represents the gray-level co-occurrence matrix (GLCM) feature extracted from the image I'. The summation ensures that multiple filters and co-occurrence features are aggregated to compute the MST feature F_T. Then F_s and F_T features are integrated to form deep texture visual features F_{DT}. In parallel, semantic features are extracted from textual class descriptions using multiple language models: fastText, BERT, and GloVe, which is shown as:

$$F_L = f_{\text{LM}}(\text{Input Label}) \tag{4}$$

where F_L represents the extracted semantic feature vector and F_{LM} denotes the semantic embedding model used to generate the embedding from the input class label.

The final feature representation F_{final} is formed by merging deep-texture features. F_{DT} and semantic features F_L given by:

$$F_{\text{final}} = [F_{DT}, F_L] \tag{5}$$

Sparse categorical cross-entropy loss is used to train the deep spatial-texture ResNet model. Once trained, the model is utilized for ZSIC, where an unseen class is predicted based on the closest feature representation.

2.2 Visual feature extraction

Visual attributes play a vital role in image recognition, making feature extraction essential. In the proposed method, DS features are obtained using a pretrained ResNet-50 model to capture high-level spatial representations. To further enhance textural representation, MST features are incorporated by aggregating GLCM and Gabor features as proposed in [11]. GLCM captures spatial statistical properties at varying distances and angles, providing information about local texture patterns, such as smoothness, roughness, and regularity. In contrast, Gabor filters capture directional frequency-based textures at different scales and orientations, enabling the extraction of edge and texture details. Since Gabor filters and GLCM descriptors provide complementary texture information, their combination yields a more discriminative and comprehensive texture representation of visual content.

2.3 Semantic feature extraction

Semantic features are high-level, attribute-based textual representations that describe class characteristics in ZSL. These features link seen and unseen classes, enabling the model to generalize to new categories without direct training samples. In ZSL, semantic features are typically derived from attribute descriptions, word embeddings, and graph-based representations. Attribute descriptions refer to manually defined characteristics (e.g., a zebra has stripes, whereas a horse does not). Graph-based representations utilize knowledge graphs or concept hierarchies (e.g., WordNet) to define class relationships. Word embeddings, often referred to as word vectors, depict words as numerical coordinates within a high-dimensional feature space.

In this arrangement, terms that share similar meanings are in proximity to one another. Word-embedding techniques are generally grouped into distributional and context-driven models. The proposed method introduces several semantic embedding models, including fastText, GloVe, BERT, and MPNet-based on sentence-transformers. All these models transform text labels into numerical embedding vectors of various dimensions (e.g., 50, 100, 200, 300) depending on the level of semantic information encoded. Let $L = \{w_1, w_2, \ldots\ldots, w_n\}$ be the set of words in a class label. Each word w_i is mapped to a d -dimensional embedding vector $e(w_i)\, \epsilon\, \mathbb{R}^d$ by the embedding model, as specified by the equation below:

$$\mathbf{E}(L) = \frac{1}{n}\sum_{i=1}^{n}\mathbf{e}(w_i), \tag{6}$$

where E_L is the semantic representation of the label L, and d is typically 300 in this work. The proposed framework employs a 300-dimensional semantic feature vector as an efficient measure of label similarity.

(1) FastText

FastText, developed by Facebook, compactly represents words as continuous vectors that retain both syntactic and semantic characteristics. Unlike conventional embedding methods, fastText enriches representations through subword information, allowing it to capture word structure and handle rare or unseen terms effectively. This procedure can be represented as:

$$f_{\text{fastText}}(w_i) = \sum_{g \in G(w_i)} \mathbf{z}_g \tag{7}$$

where $G(w_i)$ represents the collection of character n-grams found in the word, w_i and Z_g denotes the embedding vector associated with the subword unit g.

(2) BERT

Bidirectional Encoder Representations from Transformers (BERT) is a model for contextual word embeddings that derives semantic features from the surrounding text. In contrast to conventional embedding techniques, BERT employs a bidirectional transformer architecture that interprets words in the context of their neighbors within a sentence [27]. This procedure can be represented by:

$$f_{\text{BERT}}(w_i) = \text{BERT}_\theta(w_1, w_2, \ldots, w_n)[i] \tag{8}$$

where $f_{BERT}(w_i)$ indicates the embedding of the token w_i, and $BERT_\theta$ denotes the parameters of the pre-trained BERT model. The extracted features are used to align class labels with visual features using methods such as cosine similarity. This enables predictions about unseen classes based on their semantic connections.

(3) GloVe

Global vectors for word representation (GloVe) [28] is a prominent word-embedding approach that identifies relationships between words by analyzing their frequency and distribution in large text corpora. In contrast to models that rely primarily on context, GloVe creates dense word vectors by factorizing a term co-occurrence matrix. This approach uses global co-

occurrence data rather than local context, enabling it to capture semantic connections between words [28]. Each word is mapped into a dense embedding space, in which semantically similar terms appear in proximity. The embedding process can be expressed by:

$$f_{\text{GloVe}}(w_i) = \mathbf{v}_i \in \mathbb{R}^d \tag{9}$$

where f_{GLoVe} denotes the GloVe embedding function, w_i represents the input word, and v_i refers to its corresponding learned embedding vector of dimension d. These embeddings preserve semantic and syntactic relationships, making them suitable for ZSL.

(4) MPNet-based sentence embeddings

A sentence transformer model based on the MPNet architecture generates contextualized semantic embeddings using self-attention mechanisms that capture relationships among tokens within a sequence. Given a class label or textual description, the input token sequence (T) is defined as:

$$T = \{t_1, t_2, \ldots, t_n\} \tag{10}$$

where t_i represents the i^{th} token in the sequence and n denotes the total number of tokens. Each token is first converted into an embedding vector and then passed through a series of stacked transformer encoder layers. In each encoder layer, self-attention is calculated through the scaled dot-product method outlined in the following equation:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^{\top}}{\sqrt{d_k}}\right)V \tag{11}$$

where Q, K, and V are the query, key, and value matrices derived from the token embeddings, with d_k denoting the dimensionality of the key vectors. The output of the transformer encoder (H) is a list of contextualized token representations, which is shown as:

$$H = \text{Transformer}(T), \quad H \in \mathbb{R}^{n \times d} \tag{12}$$

In this equation, $\mathbb{R}$ denotes the set of real numbers, n is the number of tokens, and d is the dimensionality of the embeddings. Mean pooling is used to generate a fixed-length semantic embedding for each class from the token-level representations, as expressed in the equation below. These semantic embedding vectors are later used as input to the post-embedding optimization module.

$$z = \frac{1}{n}\sum_{i=1}^{n} H_i \tag{13}$$

where z represents the final sentence-level semantic embedding, H denotes the contextualized embedding of the i^{th} token, and n is the total number of tokens in the input sequence.

This work uses the all-mpnet-base-v2 model from the sentence transformer family to extract high-quality sentence-level embeddings. This model is built upon the standard transformer architecture and has been fine-tuned specifically for high-quality sentence-level representations via contrastive learning objectives.

2.4 Optimized autoencoder driven semantic feature extraction (*OADSFE*) *for ZSIC*

The OADSFE framework extends HFA by introducing semantic feature enhancement and optimization within the semantic feature extraction process. As illustrated in Fig. 2, the framework consists of two primary processing streams, namely visual feature extraction and semantic feature extraction. For visual features, the input image undergoes preprocessing steps

such as resizing and normalization. DS features are extracted using ResNet-50, while MST features are derived using Gabor filters and GLCM. These visual features are then integrated to form the deep-texture feature representation F_{DT}, as defined in the HFA. Semantic features F_L are extracted using a language model.

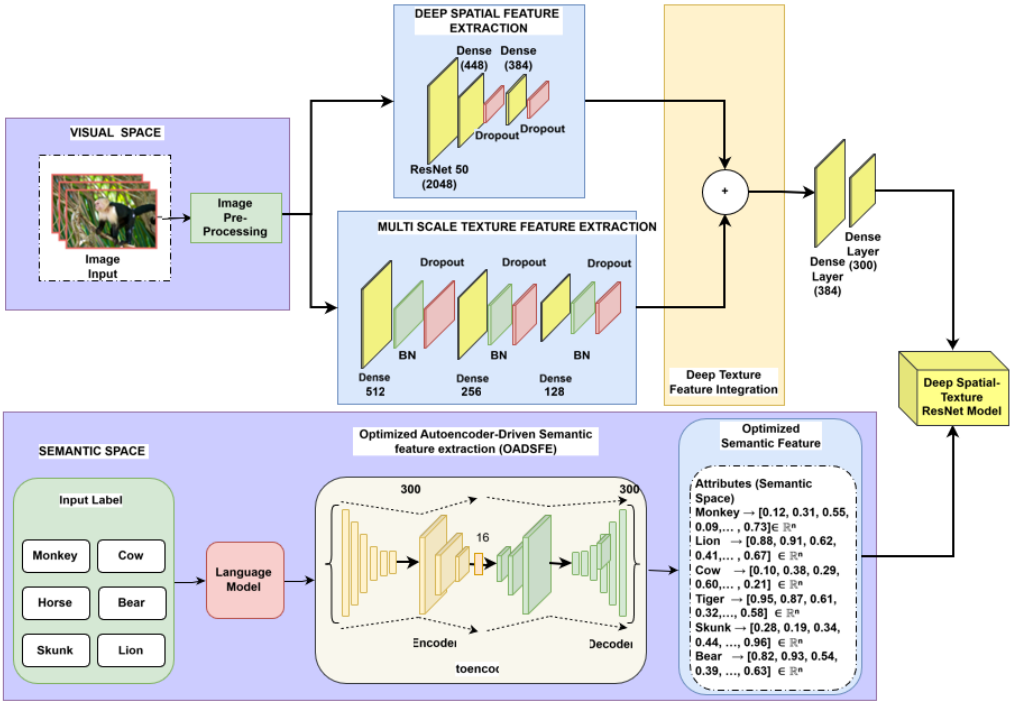


Fig. 2 Proposed optimized autoencoder-driven semantic feature extraction (OADSFE) for ZSIC

In the proposed OADSFE framework, semantic embeddings from different pre-trained language models are optimized using an autoencoder and mapped to a common latent space. The autoencoder-based module performs post-embedding optimization before classification. Specifically, FastText, GloVe, and BERT embeddings of 300 dimensions, and MPNet-based sentence transformer of 768 dimensions are all compressed into a unified 16-dimensional latent representation. This dimensional normalization enables the framework to support multiple embedding models in a scalable manner while substantially reducing memory usage and computational complexity during inference. Moreover, the autoencoder in OADSFE not only optimizes features by reducing dimensionality but also retains the most informative ones. The word embedding of an input label is first computed and then compressed and refined through an autoencoder, as described below:

$$F_L^{opt} = f_{\text{Autoencoder}}(F_L) \tag{14}$$

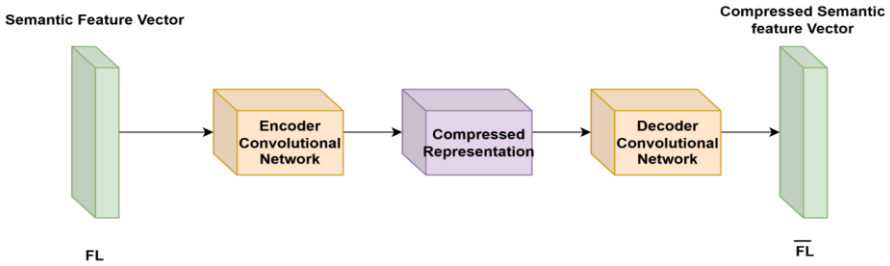
where F_L^{opt} represents the optimized semantic feature vector, F_L denotes the original semantic embedding obtained from the embedding model, and $f_{Autoencoder}$ represents the autoencoder-based semantic optimization function. This optimization process suppresses redundant information and improves feature quality. The proposed deep spatial-texture ResNet model learns a mapping from the deep-texture visual feature representation to the optimized semantic feature space, as expressed below:

$$F_L^{opt} = \Phi_{\text{ResNet}}(F_{DT}) \tag{15}$$

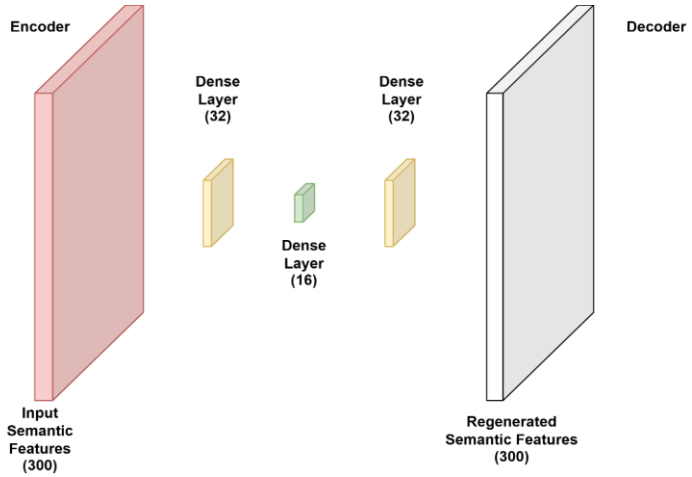
where $\Phi_{ResNet}(.)$ denotes the mapping learned by the proposed deep spatial-texture ResNet model, F_{DT} represents the deep-texture visual feature vector, and F_L^{opt} corresponds to the optimized semantic feature vector. The deep network is trained on these final features. Once trained, it performs ZSIC by predicting the label of an unseen class based on the similarity between visual and semantic representations.

(1) Autoencoders

An autoencoder is primarily designed for feature learning using an encoder–decoder structure, as shown in Fig. 3(a). The process involves encoding the original features into a lower-dimensional space (a latent space) using an encoder and then decoding them back to reconstruct the original input as closely as possible using a decoder. The latent representation is expected to capture the most important information from the original features while reducing noise and redundancy. This process not only enables dimensionality reduction but also facilitates latent feature learning, as the learned latent representations are often more compact, efficient, and discriminative than the original features.



(a) Block diagram of autoencoder



(b) Proposed autoencoder for OADSFE for ZSIC

Fig. 3 Post-embedding stage of the proposed OADSFE framework

Fig. 3(b) illustrates the proposed autoencoder for the OADSFE framework. It has a fully connected, symmetric encoder–decoder architecture. The encoder has three dense layers with 64, 32, and 16 neurons, with the 16-neuron layer representing the latent semantic space. The decoder reconstructs the input semantic features using dense layers of size 32, 64, and the original input dimension. ReLU activation functions are used in the hidden layers, while a sigmoid activation is employed in the output layer for reconstruction. A dropout rate of 0.3 is incorporated in both the encoder and decoder to reduce overfitting. The autoencoder is trained using the Adam optimizer with MSE loss, a batch size of 32, and 500 training epochs. During training, 10% of the data is used for validation. The encoder-generated latent representation is used as the optimized semantic embedding for ZSIC. The proposed autoencoder reduces the dimensionality of semantic features from the original embedding dimension (e.g., 300 or 768) to a compact 16-dimensional latent representation. The purpose of the autoencoder is to retain only the most significant features and remove redundant features.

After training, the encoder component of the autoencoder is extracted and integrated into the semantic pipeline. By combining word embeddings with an autoencoder, the proposed framework yields a more efficient and compact representation. This approach strengthens overall model performance by optimizing the semantic feature vector, which can then be aligned with visual features for applications such as ZSIC. Consequently, the model can identify new classes with semantic correspondence and speed up the classification due to reduced dimensionality.

(2) Autoencoder vs PCA for semantic dimensionality optimization

Although dimensionality reduction is traditionally performed using linear methods such as PCA, these methods are limited to linear projections. It may not adequately preserve semantic neighborhood structures in high-dimensional language embeddings. Semantic embeddings produced by models such as BERT and MPNet are highly nonlinear and encode complex contextual relationships that a linear subspace projection cannot capture.

In contrast, an autoencoder learns a nonlinear, data-driven compression that is optimized to reconstruct semantic representations while preserving discriminative relationships between classes. This allows the OADSFE framework to retain semantic similarity after dimensionality optimization, which is critical for ZSL, where class relations guide recognition of unseen categories. Accordingly, OADSFE adopts an autoencoder rather than PCA for semantic dimensionality optimization.

(3) Proposed autoencoder for OADSFE for ZSIC

The proposed OADSFE framework combines a language model and an autoencoder for optimized semantic feature extraction. Given semantic feature vectors F_L generated by a language model, an autoencoder is applied to perform dimensionality reduction and feature selection, generating Z an optimized semantic vector. Subsequently, the decoder reconstructs the semantic features Z back to the reconstructed semantic vector $\overline{F_L}$ as shown in Fig. 3(a). This optimization creates compact and meaningful representations, improves the quality of semantic features, and reduces the computational load in ZSIC. The input semantic feature vector for the autoencoder $F_L \epsilon \mathbb{R}^{300}$ corresponds to the original word embedding. The encoder compresses the input into a compressed latent representation using the equation below:

$$Z = f_{\text{Encoder}}(F_L) \tag{16}$$

where Z is the compressed semantic feature vector, and $f_{encoder}$ is the encoder function of the autoencoder. Z retains the most important information for classification. The decoder reconstructs the semantic features back to $F_L \epsilon \mathbb{R}^{300}$, as shown below:

$$\overline{F_L} = f_{\text{Decoder}}(Z) \tag{17}$$

where $\overline{F_L}$ is the reconstructed semantic vector and $F_{decoder}$ is the decoder function of the autoencoder. The autoencoder is optimized by minimizing the mean squared error (MSE) between the input semantic features and their reconstructions, which can be calculated as:

$$\mathcal{L}_{\text{recon}} = \frac{1}{n}\sum_{i=1}^{n}\left\|\mathbf{F}_L^{(i)} - \overline{\mathbf{F}}_L^{(i)}\right\|^2 \tag{18}$$

where $\mathcal{L}_{recon}$ denotes the reconstruction loss computed using MSE, $\overline{F_L^i}$ represents the original semantic feature vector of the i^{th} sample, F_L^i denotes the reconstructed semantic feature vector, and n is the total number of semantic feature samples used in the reconstruction process.

3. Results and Analysis

The OADSFE framework developed for ZSIC improves visual and semantic features through semantic feature optimization. To evaluate its effectiveness, experiments were conducted on three standard datasets: AWA2, SUN, and CUB. The SUN dataset comprises 14,340 scene images across 717 categories, of which 645 categories are used for training and 72 categories for testing. The AWA2 dataset comprises 37,322 animal images distributed across 50 classes, with 40 classes for training and 10 classes for testing. The CUB dataset contains 11,788 bird images spanning 200 species, including 150 for training and 50 for testing.

All experiments were conducted on a Windows 11–based laptop equipped with a 13th Gen Intel® Core™ i7-1355U processor (1.70 GHz), 16 GB RAM, and Intel® UHD integrated graphics. The same hardware configuration was used consistently for all accuracy, testing time, and memory consumption evaluations.

3.1 Selection of latent space dimensionality

The dimensionality of the semantic latent space plays a crucial role in balancing representation compactness and information preservation. To determine the optimal semantic compression level in OADSFE, the effect of the autoencoder's latent dimensionality was systematically evaluated. The latent dimension varied from 32 to 4, and the corresponding training and validation reconstruction losses were recorded. Fig. 4 illustrates the relationship between semantic latent dimensionality and reconstruction quality. As shown in Fig. 4, both training and validation losses attain their minimum values when the latent dimension is set to 16, indicating the best balance between information preservation and redundancy reduction. Accordingly, a 16-dimensional latent space was selected for all subsequent experiments.

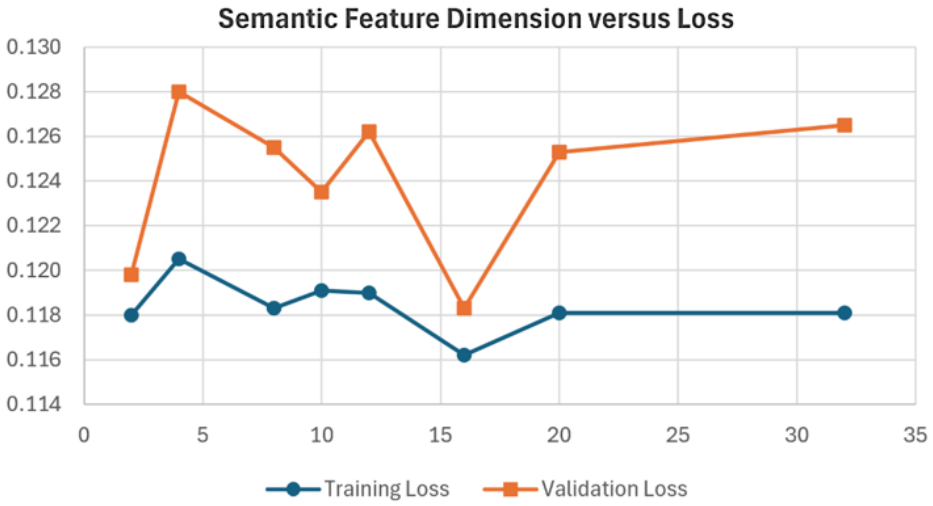


Fig. 4 Training and validation losses for different latent dimensions of the proposed autoencoder

3.2 Accuracy

The classification performance is evaluated using average per-class accuracy, which is obtained by evaluating accuracy for every class separately and averaging the results, as defined by the equations below:

$$\text{Accuracy}_i = \frac{C_i}{N_i} \times 100\% \tag{19}$$

$$\text{Accuracy}_{\text{avg,class}} = \frac{1}{C}\sum_{i=1}^{C} \text{Accuracy}_i \tag{20}$$

where C_i is the number of correct predictions for the class i.

The performance of the proposed HFA approach, with and without OADSFE, on the SUN, AWA2, and CUB datasets is summarized in Table 2. The results indicate that semantic dimensionality optimization slightly reduces accuracy on SUN and CUB, where semantically similar and fine-grained classes make discrimination more challenging. In contrast, improved performance is observed on AWA2 across several semantic embeddings, including fastText, BERT, and MPNet, which may be attributed to a greater number of semantically distinct classes. Among the proposed semantic embedding variants, MPNet-based Sentence Transformer embeddings achieved higher accuracy, particularly on AWA2, whereas BERT-based embeddings demonstrated relatively stable performance across datasets. Overall, the findings indicate a trade-off between classification accuracy and computational efficiency (testing time and memory) resulting from semantic dimensionality optimization.

Table 2 Comparison of existing ZSL methods and the proposed HFA variants with and without OADSFE

Sr No.	ZSL Approaches	SUN	AWA2	CUB
1.	DEM [2]	61.9	67.1	51.7
2.	CVAE [3]	61.7	65.8	52.1
3.	HFM [4]	53.8	65.5	69.5
4.	JG-ZSL [5]	60.3	69.4	52.9
5.	E-VAEGAN [6]	65.3	71.1	65.1
6.	DeViSe [7]	56.5	59.7	52.0
7.	ESZSL [8]	54.5	58.6	53.9
8.	LATEM [9]	55.3	55.8	49.3
9.	EZSL [10]	39.5	44.7	53.9
10.	HRT [13]	63.9	67.3	71.7
11.	ZSLViT [14]	68.3	70.7	78.9
12.	SAE [18]	40.3	54.1	33.3
13.	LESAE [21]	60.0	68.4	53.9
14.	SRSA [22]	64.3	68.3	59.9
15.	SYNC [29]	56.3	46.6	55.6
Proposed Methods and Variants				
16.	HFA for ZSIC (fastText)	62.3	62.15	53.1
17.	HFA for ZSIC (fastText with OADSFE)	61.7	63.06	51.5
18.	HFA for ZSIC (BERT)	61.8	70.3	46.93
19.	HFA for ZSIC (BERT with OADSFE)	60.5	71.61	45.29
20.	HFA for ZSIC (GloVe)	62.5	**75.78**	51.59
21.	HFA for ZSIC (GloVe with OADSFE)	62.8	73.5	51.24
22.	HFA for ZSIC (MPNet)	65.9	**78.09**	55.38
23.	HFA for ZSIC (MPNet with OADSFE)	64.43	**76.25**	54.54

Table 2 compares existing ZSL techniques with the proposed frameworks. The HFA model demonstrates stable performance across different semantic embedding strategies, with the MPNet-based HFA variant producing higher accuracy among the proposed configurations. The OADSFE framework further extends HFA by performing semantic dimensionality optimization before visual–semantic embedding, resulting in improved performance on AWA2 while introducing only marginal variations in accuracy on SUN and CUB. Recent transformer-based approaches, such as HRT [13] and ZSLViT [14], achieve strong performance through attention-driven visual–semantic representation learning, particularly on fine-grained datasets such as CUB. However, these approaches primarily focus on improving representation learning using computationally intensive transformer architectures. In contrast, the proposed HFA framework with OADSFE focuses on optimizing semantic features to reduce testing time (Section 3.3) and memory usage (Section 3.4) while maintaining effective classification performance, particularly on the coarse-grained AWA2 dataset.

As shown in Fig. 5, AWA2 contains only 10 unseen classes, and the semantic features are sparsely distributed and well separated in PCA space. This strong inter-class separability enables the low-dimensional embeddings to preserve discriminative boundaries effectively, thereby facilitating accurate classification.

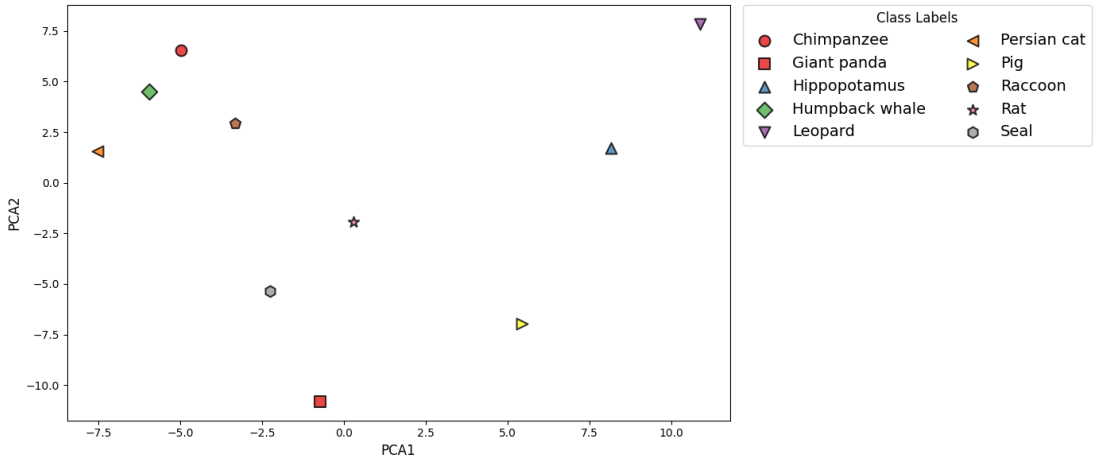


Fig. 5 Semantic feature distribution of unseen test classes in the AWA2 dataset

Fig. 6 visualizes the semantic distributions of unseen classes for the CUB dataset, with all classes shown in gray and selected semantically similar groups highlighted to emphasize local relationships. As presented in Fig. 6, semantically similar classes tend to form compact and partially overlapping clusters in the embedding space. Such overlap reduces inter-class separability, making it more challenging for the model to learn clearly distinguishable decision boundaries. As a result, this may increase the probability of misclassification for the CUB dataset.

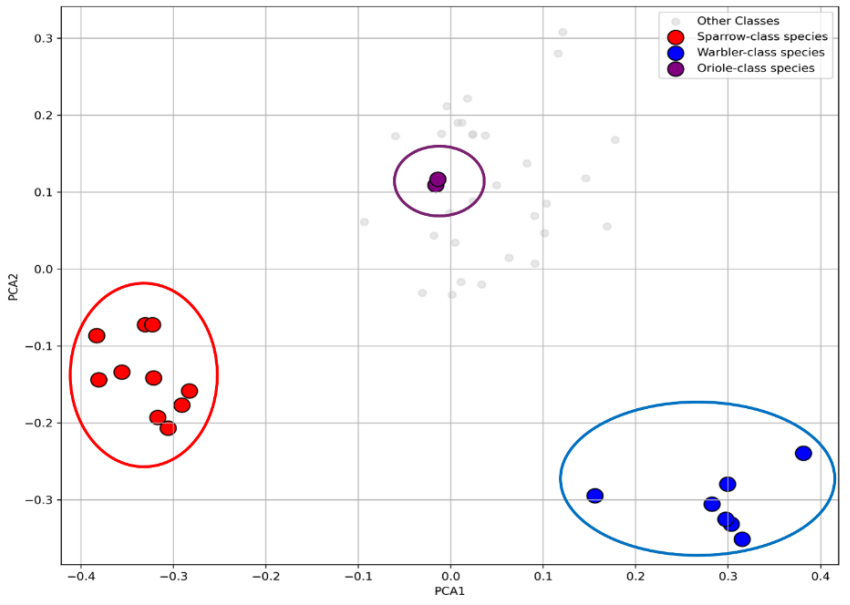


Fig. 6 Semantic feature distribution of unseen test classes in the CUB dataset

Fig. 7 illustrates the semantic distribution of unseen classes in the SUN dataset. Unlike CUB, their classes are more evenly distributed in the PCA space, indicating a greater semantic diversity. The highlighted pairs (kitchen vs. living room and forest vs. jungle vs. mountain vs. valley) are small local clusters in a very similar semantic space and not far from each other. Thus, while the semantically related categories remain close in the embedding space, different groups of scenes are well separated. Consequently, the overlap is low globally, but there is still a well-defined relationship locally for class separation and, unlike CUB, the classification complexity is moderate.

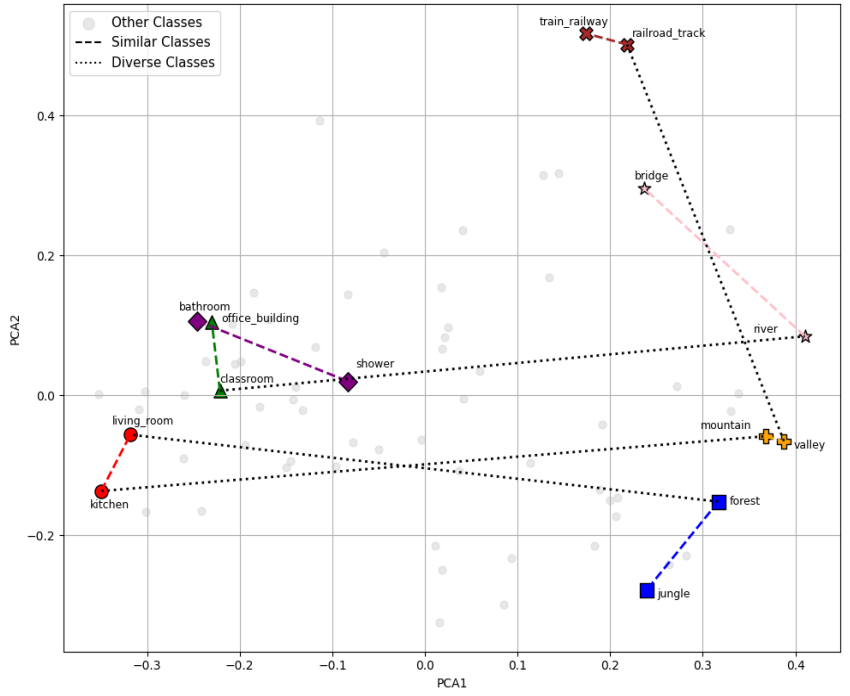


Fig. 7 Semantic feature distribution of unseen test classes in the SUN dataset

Overall, the effectiveness of the proposed method is strongly influenced by the degree of semantic separability within the dataset. Datasets with well-separated semantic structures, such as AWA2, tend to achieve higher accuracy, whereas fine-grained datasets with dense and overlapping semantic representations, such as CUB, pose greater challenges. The SUN represents an intermediate case, balancing local similarity and global separability.

3.3 Time complexity analysis

Time complexity in zero-shot inference reflects the computational cost of processing test samples and generating predictions. In this work, time complexity is evaluated empirically through the average testing time per class. The average testing time is calculated for each class, and then the mean is given by the equation below:

$$T_{\text{avg,class}} = \frac{1}{C}\sum_{i=1}^{C}\frac{T_i}{N_i} \tag{21}$$

where C is the total number of classes, N_i is the number of samples in class I, and T_i corresponds to the overall testing time of that class.

Table 3 summarizes the reduction in testing time achieved by OADSFE across different semantic embedding models. FastText exhibits the largest reduction in testing time, followed by GloVe and BERT. Although the reduction observed for MPNet is relatively smaller, semantic dimensionality optimization consistently decreases testing time for all embedding models. On AWA2 dataset, OADSFE reduces the average testing time by 16.29% across embedding models. This improvement is primarily due to the lower-dimensional optimized semantic representations, which decrease model complexity and accelerate feature processing during zero-shot inference.

Table 3 Testing time comparison (in seconds) and percentage reduction using OADSFE across datasets

Dataset	Embedding	HFA	HFA + OADSFE	Decrease (%)
AWA2	fastText	0.60	0.46	23.75
	BERT	0.54	0.46	15.37
	GloVe	0.52	0.44	16.00
	MPNet	0.53	0.48	10.03
SUN	fastText	0.92	0.70	23.91
	BERT	0.86	0.73	15.12
	GloVe	0.83	0.70	15.66
	MPNet	0.88	0.79	10.23
CUB	fastText	0.78	0.60	23.08
	BERT	0.73	0.62	15.07
	GloVe	0.70	0.59	15.71
	MPNet	0.75	0.68	9.33
Average Testing Time Reduction				16.1

3.4 Memory consumption analysis

The memory consumption of the semantic embeddings estimates the memory required to store an embedding of a given dimension. It is calculated by the equation below for an embedding of dimension. *D* applied to *N* images.

$$\text{Storage} = N \times D \times 4 \ \ \text{ytes} = O(ND) \tag{22}$$

where 4 bytes correspond to the storage size of a single floating-point number.

Table 4 shows that applying the autoencoder in the OADSFE framework significantly reduces memory requirements across all datasets. For AWA2 dataset, storage decreases from 7.54 MB to 0.57 MB, representing a 92.44% reduction. AWA2 shows the highest memory reduction (92.44%), followed by CUB (89.68%) and SUN (86.15%). Table 5 further summarizes the reduction in memory consumption achieved by OADSFE when applied to high-dimensional MPNet embeddings. Owing

to their original 768-dimensional representation, MPNet embeddings require substantially more storage than those of other language models. By compressing these embeddings into a unified 16-dimensional latent space, OADSFE reduces the average storage requirement from 10.96 MB to 0.23 MB across datasets, corresponding to a reduction of 97.94%. Among the datasets, SUN achieves the highest reduction of 97.99%, followed closely by AWA2 and CUB, each achieving 97.92%.

Table 4 Memory requirement with and without OADSFE

Dataset	HFA (MB)	HFA + OADSFE (MB)	Memory Reduction (%)
AWA2	7.54	0.57	92.44
SUN	1.95	0.27	86.15
CUB	3.39	0.35	89.68
Average	4.29	0.23	89.42

Table 5 Memory consumption of MPNet embeddings with and without OADSFE

Dataset	MPNet (768-D) (MB)	MPNet + OADSFE (16-D) (MB)	Memory Reduction (%)
AWA2	19.23	0.40	97.92
SUN	4.98	0.10	97.99
CUB	8.66	0.18	97.92
Average	10.96	0.23	97.94

These results indicate that semantic dimensionality reduction is particularly beneficial for contextualized language models, which generally require substantial memory demand. The improvements are especially notable in high-dimensional MPNet embeddings, where aggressive dimensionality reduction yields substantial storage savings while maintaining competitive classification performance. This demonstrates the scalability of OADSFE, making it appropriate for resource-constrained and large-scale ZSIC applications. Even after semantic dimensionality reduction, the embeddings retain sufficient discriminative information to achieve competitive ZSIC accuracy.

3.5 Ablation study: PCA vs autoencoder for semantic dimensionality optimization

For the autoencoder-based semantic dimensionality optimization, an ablation study is performed on AWA2 using MPNet embeddings. Three variants are evaluated:

(1) Raw MPNet embeddings (768D).

(2) PCA-compressed MPNet embeddings (768D→16D)

(3) OADSFE autoencoder-compressed MPNet embeddings (768D→16D)

Table 6 compares the classification performance of these configurations. Although the original 768-dimensional MPNet embeddings obtain the highest classification accuracy, they incur significantly higher memory and inference overhead. The PCA-based dimensionality reduction results in a noticeable drop in ZSIC performance, indicating a loss of discriminative semantic information. In contrast, the proposed OADSFE autoencoder preserves the semantic neighborhood structure and retains stronger discriminative semantic features while using the same 16-dimensional constraint, yielding improved ZSL accuracy for the compressed representations.

Table 6 ZSIC accuracy (%) on AWA2 using MPNet embeddings with multiple dimensionality optimization approaches

No.	Method	Accuracy (%)
1.	Raw embedding (768D)	78.09
2.	PCA (768D→16D)	70.45
3.	OADSFE (Autoencoder, 768D→16D)	73.41

The obtained accuracy of 73.41% demonstrates that the proposed autoencoder captures informative latent semantic characteristics from the original 768-dimensional MPNet embeddings while removing redundant information. Furthermore, the autoencoder's nonlinear encoding capability enables better preservation of inter-class semantic relationships than linear

PCA, thereby contributing to improved classification performance. These results indicate that autoencoder-driven nonlinear compression is more effective than linear PCA for optimizing semantic dimensionality.

4. Conclusion

This paper proposed an OADSFE framework to enhance ZSIC by refining semantic representations within the HFA approach. The framework integrated deep spatial and multi-scale texture visual features with semantic embeddings derived from multiple models, including fastText, GloVe, BERT, and MPNet. An autoencoder-based post-embedding semantic optimization module was employed to compress high-dimensional embeddings into compact latent representations while preserving informative semantic characteristics. The main conclusions of this study are summarized as follows:

(1) Embedding model evaluation: Various embedding models are evaluated independently, enabling identification of the most effective semantic features for ZSL.
(2) Semantic Optimization Effectiveness: The post-embedding autoencoder in the proposed OADSFE approach successfully reduces semantic dimensionality while maintaining discriminative capability.
(3) Improved Efficiency: The proposed approach achieves up to 16.29% reduction in testing time and 89.42% reduction in memory usage compared with the baseline HFA framework under the same hardware and experimental settings.
(4) Balanced Accuracy–Efficiency Trade-off: The framework effectively balances classification performance with memory consumption and testing time, indicating its potential suitability for resource-constrained and large-scale ZSL scenarios.
(5) General Applicability: The proposed approach demonstrates robustness across diverse ZSL scenarios and is therefore adaptable to different semantic embedding models and datasets.

Overall, the results demonstrate that OADSFE can improve semantic compactness while maintaining effective ZSIC performance with reduced computational overhead. Future work may explore incorporating more advanced transformer-based semantic representations and hybrid generative embedding strategies to further enhance fine-grained recognition performance.

Conflicts of Interest

The authors declare no conflict of interest.

References

[1] G. Ramesh, M. Sahil, S. A. Palan, D. Bhandary, T. A. Ashok, J. Shreyas, et al., “A Review on NLP Zero-Shot and Few-Shot Learning: Methods and Applications,” Discover Applied Sciences, vol. 7, no. 9, article no. 966, 2025.
[2] L. Zhang, T. Xiang, and S. Gong, “Learning a Deep Embedding Model for Zero-Shot Learning,” Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, pp. 3010-3019, 2017.
[3] A. Mishra, S. Krishna Reddy, A. Mittal, and H. A. Murthy, “A Generative Model for Zero Shot Learning Using Conditional Variational Autoencoders,” Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), IEEE, pp. 2188-2196, 2018.
[4] F. Al Machot, M. Ullah, and H. Ullah, “HFM: A Hybrid Feature Model Based on Conditional Auto Encoders for Zero-Shot Learning,” Journal of Imaging, vol. 8, no. 6, article no. 171, 2022.
[5] M. Zhang, X. Wang, Y. Shi, S. Ren, and W. Wang, “Zero-Shot Learning with Joint Generative Adversarial Networks,” Electronics, vol. 12, no. 10, article no. 2308, 2023.
[6] B. Ding, Y. Fan, Y. He, and J. Zhao, “Enhanced VAEGAN: A Zero-Shot Image Classification Method,” Applied Intelligence, vol. 53, no. 8, pp. 9235-9246, 2023.
[7] A. Frome, G. S. Corrado, J. Shlens, S. Bengio, J. Dean, M. A. Ranzato, et al., “DeViSE: A Deep Visual-Semantic Embedding Model,” Proceedings of the 27th International Conference on Neural Information Processing Systems, pp. 2121-2129, 2013.
[8] B. Romera-Paredes and P. Torr, “An Embarrassingly Simple Approach to Zero-Shot Learning,” Proceedings of the International Conference on Machine Learning (ICML), PMLR, pp. 2152-2161, 2015.

[9] 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 (CVPR), IEEE, pp. 69-77, 2016.

[10] A. S. Khanam and P. N. Sonar, "Enhanced Zero Shot Learning Using Deep Neural Network ResNet50," Proceedings of the 2023 4th International Conference for Emerging Technology (INCET), IEEE, pp. 1-6, 2023.

[11] S. Khanam and P. N. Sonar, "Hybrid Feature Approach for Enhancing Zero-Shot Image Classification," in Artificial Intelligence and Knowledge Processing, H. Hemachandran, R. V. Rodriguez, M. Rege, A. Ade-Ibijola, K.-L. Ong, and V. Piuri, Eds., Cham: Springer Nature Switzerland, pp. 239-251, 2025.

[12] C. Wang, P. Nulty, and D. Lillis, "A Comparative Study on Word Embeddings in Deep Learning for Text Classification," Artificial Intelligence Review, vol. 55, no. 2, pp. 1501-1541, 2022.

[13] D. Cheng, G. Wang, B. Wang, Q. Zhang, J. Han, and D. Zhang, "Hybrid Routing Transformer for Zero-Shot Learning," Pattern Recognition, vol. 137, article no. 109270, 2023.

[14] S. Chen, W. Hou, S. Khan, and F. S. Khan, "Progressive Semantic-Guided Vision Transformer for Zero-Shot Learning," Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, pp. 23964-23974, 2024.

[15] Y. Palagummi and S. Rowlands, "Shifted Window Based Self-Attention via Swin Transformer for Zero-Shot Learning," International Journal of Computer and Information Engineering, vol. 17, no. 10, pp. 524-531, 2023.

[16] F. Alamri and A. Dutta, "Multi-Head Self-Attention via Vision Transformer for Zero-Shot Learning," Proceedings of the Irish Machine Vision and Image Processing Conference (IMVIP), 2021.

[17] K. Berahmand, F. Daneshfar, E. S. Salehi, Y. Li, and Y. Xu, "Autoencoders and Their Applications in Machine Learning: A Survey," Artificial Intelligence Review, vol. 57, article no. 28, 2024.

[18] E. Kodirov, T. Xiang, and S. Gong, "Semantic Autoencoder for Zero-Shot Learning," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, pp. 4447-4456, 2017.

[19] X. Zhang, Y. Zhang, and F. Shen, "Bi-shifting Semantic Auto-Encoder for Zero-Shot Learning," Knowledge-Based Systems, vol. 244, article no. 108531, 2022.

[20] J. Li, C. Chen, and W. Liu, "Zero-Shot Learning via Discriminative Dual Semantic Auto-Encoder," Neurocomputing, vol. 417, pp. 117–126, 2020.

[21] Y. Liu, Q. Gao, J. Li, J. Han, and L. Shao, "Zero Shot Learning via Low-Rank Embedded Semantic Autoencoder," Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI), AAAI Press, pp. 2490–2496, 2018.

[22] 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, article no. 108237, 2022.

[23] W. Heyden, H. Ullah, M. S. Siddiqui, and F. Al Machot, "An Integral Projection-Based Semantic Autoencoder for Zero-Shot Learning," IEEE Access, vol. 11, pp. 85351-85360, 2023.

[24] G. Patterson and J. Hays, "SUN Attribute Database: Discovering, Annotating, and Recognizing Scene Attributes," https://cs.brown.edu/~gmpatter/sunattributes.html, accessed in 2024.

[25] Y. Xian, C. H. Lampert, B. Schiele, and Z. Akata, "Animals with Attributes 2: A Free Dataset for Attribute-Based Classification and Zero-Shot Learning," https://cvml.ista.ac.at/AwA2/, 2018.

[26] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, "The Caltech-UCSD Birds-200-2011 Dataset," California Institute of Technology, Technical Report CNS-TR-2011-001, Pasadena, CA, USA, 2011.

[27] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT), pp. 4171-4186, 2019.

[28] J. Pennington, R. Socher, and C. D. Manning, "GloVe: Global Vectors for Word Representation," Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1532-1543, 2014.

[29] S. Changpinyo, W. L. Chao, B. Gong, and F. Sha, "Synthesized Classifiers for Zero-Shot Learning," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE Press, pp. 5327-5336, 2016.