Kernelized Distance Learning for Zero-Shot Recognition

Mohammad Reza Zarei[a], Mohammad Taheri[a,*], Yang Long[b]

[a]*Department of Computer Science and Engineering, Shiraz University, Shiraz, Iran*
[b]*Department of Computer Science, Durham University, Durham, UK*

Abstract

Zero-Shot Learning (ZSL) has gained growing attention over the past few years mostly because it provides a significant scalability to recognition models for classifying instances from new unobserved classes. This scalability is achieved by providing semantic information about new classes, which could be obtained remarkably easier with lower cost, compared to collecting a new training set. Because seen and unseen classes are completely disjoint, ZSL methods often suffer from domain shift problem that occurs in transferring the knowledge of seen classes to unseen ones. Moreover, hubness problem that usually arises in high-dimensional space is another challenge in most ZSL methods due to applying nearest neighbor search for classification. To address these issues, a kernelized distance function is learned in order to discriminate the classes with a customized large-margin loss function. Furthermore, a simple theoretical-based prototype learning approach is provided by defining a non-linear mapping function to learn the visual prototype of each class from associated semantic information. For classification task, the learned distance function is utilized to measure the distance between instances and class-related prototypes. The evaluation on five benchmarks demonstrates the superiority of the proposed method over the state-of-the-art approaches in both zero-shot and generalized zero-shot learning problems.

Keywords: Zero-Shot Learning, Distance Learning, Large-Margin, Prototype Learning

1. Introduction

With daily increase in novel categories of objects in nature, acquiring recognition models with the possibility of classifying instances from new classes is necessary. With traditional models, classification task highly depends on accessing sufficient instances of each class for training the model. However, this is not always attainable for new categories. While collecting sufficient data for all new classes might be difficult, the collected data should also be annotated that is an expensive task. Furthermore, training the model on the instances of all new classes is time-consuming and sometimes impractical. By considering these challenges, a new problem known as Zero-Shot Learning (ZSL) has

*Corresponding author
Email addresses: mr.zarei@cse.shirazu.ac.ir (Mohammad Reza Zarei), motaheri@shirazu.ac.ir (Mohammad Taheri), yang.long@durham.ac.uk (Yang Long)

been introduced, which focuses on recognizing instances from classes not covered in the training stage [1].

ZSL utilizes semantic information to achieve some relations between training classes (seen classes) and test classes (unseen classes) and fill the gap between them. This information could be derived in different manners, such as semantic attributes provided by experts to characterize visual and semantic properties of objects [1, 2], or vector embedding of class names learned from a text corpus through unsupervised modeling [3]. Regardless of the type of utilized semantic information, each class is presented with a single vector of features in semantic space. This space differs from the visual feature space in which all instances from different classes are presented. While only instances of seen classes are accessible in the training stage, semantic vectors of both seen and unseen classes are always available. ZSL methods aim to discover some relations between classes in the semantic space. In other words, the links between seen and unseen classes are attempted to be discovered in the semantic space to pave the way for recognizing probable instances of unseen classes in visual feature space.

Conventional ZSL assumes that instances from only unseen classes will be provided to the model for classification in the testing stage. Therefore, the search is performed only in unseen classes. However, it is more reasonable and realistic to expect observing instances from both seen and unseen classes during the testing phase. Hence, the true class should be searched among all seen and unseen classes. This testing scheme is often called Generalized Zero-Shot Learning (GZSL) [4].

From the perspective of methodology, ZSL models can be roughly grouped into three categories: (1) global compatibility learners (2) space unifier methods (3) generative methods. ZSL in global compatibility learner methods [5, 6, 7] is achieved by learning a global compatibility function between the visual features and semantic vectors such that the semantic vector of correct (seen) class attains the highest compatibility score with each training instance. Space unifier methods project either visual features or semantic vectors or both to a predefined embedding space. Then, the nearest neighbor search is applied to recognize the class of a new instance during the testing phase. Hence, in these methods, learning the mapping parameters can also be considered as a part of learning a proper metric for distance estimation. A line of research aims for the semantic space as the target of mapping instances [8, 9] to preserve the semantic structure. Conversely, some approaches map the semantic features to the visual feature space [10, 11, 12] resulting in class-specific visual prototypes. The last trend in space unifier methods is to map both visual features and corresponding semantic vectors into a third space by defining an objective function to align them [13]. In generative methods as the last category of ZSL [14, 15, 16], training instances of seen classes are analyzed to learn resampling for each class using the associated semantic vector. Then sufficient instances for unseen classes are generated to convert zero-shot problem to a conventional classification task. Generative networks, e.g., Generative Adversarial Network (GAN) [17, 18, 19], are usually utilized to accomplish this objective. These models are different from the previous two categories in architecture with more time and memory complexity but less visibility and interpretability. This paper focuses on space unifier methods.

One of the main challenges in ZSL is the hubness problem [20]. Hubness that usually occurs in high-dimensional spaces is the existence of instances from different classes (hubs) in the neighborhood of test instances that belong to the same category [21]. Space unifier methods especially those that target the semantic space, suffer from the hubness

problem due to utilizing nearest neighbor search in classification phase. To alleviate this issue, various methods attempted to learn a proper metric instead of utilizing common general distances such as Euclidean distance [12, 22]. However, disjoint training and testing classes still deteriorate the performance of the model in the testing stage.

Although there are some similarities between seen and unseen categories that bring the possibility of zero-shot recognition, the evident difference between data distributions of source (training) and target (test) domains causes a bias towards seen classes which prevents the model from achieving a reasonable recognition performance [23]. This phenomenon that ZSL methods struggle with is called the domain shift problem [24]. To address such issue, instead of considering ZSL with inductive setting, a line of research attempts to employ unlabeled data of unseen classes in addition to labeled instances of seen classes in the training stage which is called transductive ZSL [24]. However, the data of unseen categories are not usually available during the training phase in realistic recognition scenarios. Therefore, transductive setting is not often considered to be the best solution for the domain shift problem.

In order to solve the aforementioned problems, in this paper, a kernelized Euclidean-based distance function is proposed in visual feature space to optimize the distance between visual features of instances and class-specific prototypes. The proposed distance function can be adopted with any prototype learning method. However, a non-linear but straitforward approach is used to learn visual prototypes from their corresponding semantic vectors. As mentioned previously, although distance learning alleviates the hubness problem in the context of ZSL, it often suffers from the domain shift problem. Kernelization has shown promising results in improving model generalization capability but is rarely studied in ZSL and GZSL contexts. Furthermore, a large-margin objective function is utilized for learning the distance function to enhance the discriminative properties of the model and reducing the impact of hubness and domain shift problems. The proposed method is evaluated by experimenting on five standard ZSL datasets under both ZSL and GZSL settings. Extensive experimental results show the effectiveness of the proposed method under both settings compared to state-of-the-art methods with various approaches. The contributions of this paper are as follows:

- A kernelized Euclidean-based distance function is proposed in visual feature space for tuning the distance of visual features with class-related prototypes in order to mitigate domain shift and hubness problems. The proposed distance function can be used with any visual prototype learning method.
- A large-margin objective is employed for learning the distance function to improve discriminative properties.
- A simple non-linear prototype learning method is used to transfer the knowledge for unseen classes from semantic space to visual feature space and construct the visual prototypes.
- Extensive experiments are conducted on five widely used ZSL datasets under ZSL and GZSL settings, and it is demonstrated that the proposed method outperforms state-of-the-art approaches.

The rest of the paper is organized as follows. In Section 2, an overview of the relevant previous work is provided. The proposed method is explained in Section 3 and Section

4 is dedicated to illustrating the experimental results. Finally, the paper is concluded in Section 5.

2. Related Work

This section investigates a set of related work in terms of prototype learning, distance function learning, large-margin loss functions, and non-linear methods.

2.1. Prototype Learning

In recent years, various ZSL methods have targeted visual prototype learning for unseen classes. Among these methods, a line of research considers minimizing "inner class distance"; denoting the mean square of the distances between training instances and associated prototypes in this paper; as the objective function. The article [11] argues that the visual feature space is a more appropriate embedding space compared to semantic space due to less impact of the hubness problem on nearest neighbor search in visual feature space. Then, it proposes a non-linear function to map semantic vector of each seen class to a proper visual prototype in the visual feature space. This mapping function is learned by a two-layer neural network with Rectified Linear Unit (ReLU) firing functions. The inner class distance is regularized with sparsity terms on the network's weights to form the final objective function. In [25], at first, the visual prototypes of seen classes are learned by minimizing the inner class distance that results in obtaining the mean of instances for each seen class as the visual prototype. Then, an embedding function is learned to map the semantic vectors to corresponding class prototypes. In order to achieve generalization, the mapping problem is solved by Support Vector Regressor (SVR). To be more precise, a separate SVR is used to predict each feature of the prototype from semantic features. In order to reduce the number of SVRs, the number of visual features is initially reduced by PCA.

In two previously mentioned methods, in addition to the type of mapping function, the difference is in the direction of problem-solving. In [25], at first, the prototypes of the seen classes are specified by minimizing inner class distance. Then, for obtaining the prototypes of unseen classes from corresponding semantic vectors, an embedding function is learned using the semantic vectors and prototypes of seen classes. While in [11], the prototypes are not fixed and the weights of the network are learned to obtain prototypes which minimize the inner distance of seen classes. In this paper, the approaches in [25], and [11] are called late and early learning, respectively. In Section 3, as one of the contributions of this paper, it is proved that these two learning methods are the same for linear mappings with inner class distance objective function.

Similarly, the article [26] employs the minimization of inner class distance to obtain the visual prototypes of seen classes. This paper assumes a set of latent vectors in a third space from which both visual prototypes and semantic vectors can be generated separately. Square errors of these mappings are also considered in the objective function in addition to inner class distance.

In this paper, similar to the mentioned methods, a visual prototype learning approach with inner class distance as the loss function is utilized to learn class-related prototypes. However, this objective does not consider the discriminative properties of learned prototypes. Therefore, instead of using a simple distance function in nearest neighbor classifier,

a large-margin objective function is used to learn a discriminative kernel-based distance function.

2.2. Distance Function Learning

To enhance the performance of ZSL, distance learning has been recently considered as an effective method rather than using the traditional distance functions such as Euclidian distance. In space unifier methods that are based on learning one or more mapping functions, utilizing proper distance metrics can lead to better dispersion of instances of different classes and relieve the hubness problem. In [27], Mensink et al. propose a Mahalanobis metric learning in the form of Gaussian distributions with class specific means and a common covariance matrix shared between all the class labels. The parameters of the distributions are learned by maximizing the log-likelihood of correct predictions. While One-Shot-Learning was considered in that work, a ZSL model based on class-specific Gaussian distributions is proposed in [10]. The difference of this model with [27] is that instead of using a shared covariance matrix, each distribution has its own specific covariance matrix to be learned. The parameters for the distributions of seen classes are learned similarly by log-likelihood maximization. Then, two distinct Kernel Ridge Regression models are trained to map each semantic class vector to its class-related mean and covariance. In the end, these regressors are used to estimate the statistics of unseen classes (means and covariance matrices). Using a specific covariance for each class can be interpreted as considering a distinct distance function for measuring the distance between an arbitrary instance and the corresponding class prototype.

In [22], Bucher et al. formulate ZSL as a Euclidean-based metric learning problem in semantic space and learn a distance function to better predict the consistency of an embedded image with its relevant semantic vector. To embed visual features, a ReLU-type normalized affine function is learned with a least-square objective function. Also, the distance function is adjusted using a Hinge loss. Although the parameters of the mapping and distance functions have different learning objectives, a joint learning approach is employed by merging these objectives.

Comparing [27] and [10] with [22], the former methods aim at finding the proper distance function between the training instances and the prototypes in the visual feature space, whereas the latter learns a distance function between the instances mapped to the semantic space, and the semantic vectors of associated class labels.

In recent years, other distance functions have also been learned besides Mahalanobis and Euclidean-based metrics. For example, in [12], a distance function based on Cosine similarity is learned between class-specific prototypes and instances in visual feature space. In this method, a two-layer neural network is used for learning visual prototypes from semantic class vectors.

Although previous methods have utilized distance function learning to relieve hubness and domain shift problems in ZSL and GZSL, they still suffer from the domain shift problem due to the inconsistent distribution of unseen classes. In this paper, kernelization of the distance function in visual feature space is one of the main contributions. While kernel-based methods can improve model generalization, they have been rarely used for ZSL and GZSL. Moreover, a large-margin loss has been employed in learning the distance function to emphasize on tuning structural and empirical risks, simultaneously. Although a large-margin distance learning approach is used in [22], it targets the semantic space that is affected by the hubness problem more than the visual feature space.

2.3. Large-Margin Loss Functions

Regardless of the type of methodology employed by ZSL models, each method uses a specific loss function to evaluate the error of categorizing training instances. Minimizing the sum of the losses on all training instances as the empirical risk is usually used for parameter learning. While mean square distance is widely used in ZSL methods, including prototype learning approaches described in subsection 2.1 [11, 25], discriminative properties of distinct classes are not usually considered in objective functions based on this distance. Therefore, it can deteriorate the generalization capability of ZSL methods. To enhance the ability of discriminating classes, large-margin objective functions have been utilized in ZSL methods. Improving this capability in ZSL can lead to mitigating domain shift and hubness problems.

In ZSL methods, large-margin loss functions are used mainly for global compatibility learners. In [6], inspired by unregularized ranking SVM [28], a pairwise ranking objective based on Hinge loss is used to tune a bi-linear compatibility function. In that method, all the opposing classes are considered in the loss function. Akata et al. [7] used a weighted objective function which also considers the rank of the correct label among all labels in the objective function. In [29] another large-margin objective function is employed that gives the whole weight to the most similar opposing class. In order to tackle the domain shift problem, that exists between seen and unseen classes, an intuitive idea was used in [5] to consider similar seen class instances as a substitution of unseen ones in the training phase. Hence, for each instance, the most similar unseen class is also considered in a loss function similar to the one used in [29] to maximize the compatibility of the instance with the most similar unseen class.

In space unifier methods, large-margin loss functions are also utilized. In [22], Hinge loss was used to define a loss function but on the distance (not compatibility) function. This method focuses on semantic space, in contrary with the proposed method, in order to discriminate the classes by tuning the margin between each mapped instance and its class-related semantic vector.

While large-margin discriminative loss functions have been primarily used for global compatibility learners in ZSL (as introduced previously), this paper attempts to utilize a large-margin loss for learning the proposed kernel-based distance function in visual feature space. This type of loss functions, based on Hinge loss has not been considered properly so far, based on our knowledge, for the goal of distance function learning, especially in visual feature space.

2.4. Non-linear Methods

In ZSL and GZSL, deep learning has been widely used to achieve non-linearity in the models [30, 11, 31]. However, these methods are less efficient and interpretable and usually more complex compared to non-deep approaches. Unlike previous methods, in [32], non-linearity is acquired by kernel utilization similar to the approach proposed in this paper. However, unlike the proposed method that targets visual feature space, semantic space was used in [32] which is not appropriate due to the severe hubness problem that exists in this space. Moreover, that method applies kernels for learning a mapping function while the proposed method uses them in distance function.

To sum up, the proposed ZSL method attempts to learn a kernel-based distance function with a large-margin objective to tune the distance between the instances and class-related prototypes in visual feature space. Although the visual prototypes are learned

Table 1: Comparison between the proposed method and related work based on various criteria.

Method	Category	Target space	Focus	Early/ Late learning	Large-margin	Kernel
Zhang et al. [11]	space unifier	visual	distance	early	no	no
Changpinyo et al. [25]	space unifier	visual	distance	late	no	yes
Jiang et al. [26]	space unifier	visual/semantic/third	distance	early	no	no
Verma and Rai [10]	space unifier	visual	distance	late	no	no
Bucher et al. [22]	space unifier	semantic	distance	-	yes	no
Pan et al. [12]	space unifier	visual	distance	early	no	no
Frome et al. [6]	compatibility learner	-	similarity	-	yes	no
Akata et al. [7]	compatibility learner	-	similarity	-	yes	no
Akata et al. [29]	compatibility learner	-	similarity	-	yes	no
Zhang and Koniusz [32]	space unifier	semantic	similarity	-	yes	yes
Proposed	space unifier	visual	distance	late	yes	yes

with a simple non-linear method, the distance function can be used with any visual prototype learning method. The proposed distance function is Euclidean-based and the kernel transformation can occur whenever the Euclidean distance function is employed. However, the kernel utilization can also be applied to any other distance function that can benefit from transforming to a Hilbert Space with the appearance of dot product of vectors. What distinguishes the proposed distance function is its kernel utilization in distance function formulation and the large-margin objective function, which is hardly manipulated previously for the goal of distance function learning, especially in visual feature space. To the best of our knowledge, the only work in the literature, based on both Kernel and Large-margin approaches, is Zhang [32] that uses a similarity function in the semantic space whereas, this paper proposes a distance function in visual feature space to prevent the hubness problem. In Table 1, the proposed method is compared to related work based on various criteria.

3. Proposed Method

In this section, the proposed method is explained, mainly focused on learning a kernelized distance function and a mapping function for visual prototype learning

Let $C^s = \{c_1^s, c_2^s, \ldots, c_S^s\}$ and $C^u = \{c_1^u, c_2^u, \ldots, c_U^u\}$ be the sets of S seen and U unseen class labels, respectively, where $C^s \cap C^u = \emptyset$. In addition, $X = \{\vec{x}_1, \vec{x}_2, \ldots, \vec{x}_N\}$ represents the set of N observed instances from the seen classes where $\vec{x}_i \in \mathbb{R}^{D_v}$ is the vector of D_v visual features for the i^{th} training instance. The class label of $\vec{x}_i$ is denoted by $y(\vec{x}_i) = y_i \in C^s$. Moreover, each class label $c \in C^s \cup C^u$ is associated with $\vec{a}_c \in \mathbb{R}^{D_s}$ as the D_s-dimensional semantic features of the seen or unseen class c. The goal of ZSL is to assign the correct class $y(\vec{q})$ to each new instance $\vec{q} \in \mathbb{R}^{D_v}$, where $y(\vec{q}) \in C^u$. However, in generalized Zero-Shot Learning, instances from both seen and unseen classes are available during the test phase and the search is performed in $C^s \cup C^u$.

Let $\vec{p}_{y_i}$ be the visual prototype associated with class y_i. In visual feature space, a distance function is defined to tune the distance of each instance $\vec{x}_i$ to its class-related visual prototype $\vec{p}_{y_i}$. By using the visual prototypes, the distance function in visual feature space is commonly formulated as a general distance shown in Eq. (1).

$$d_v\left(\vec{x}_i, \vec{p}_c; Q\right) = \left(\vec{x}_i - \vec{p}_c\right)^T Q \left(\vec{x}_i - \vec{p}_c\right) \tag{1}$$

where, Q is a symmetric positive semi-definite matrix. This function is employed to measure the distance between the instance $\vec{x}_i$ and the visual prototype of class c (i.e., $\vec{p}_c$). In order to minimize inner-class distance, the objective function is formulated as in Eq. (2).

$$\min_{Q, \vec{p}_{c \in C^s}} J\left(Q, \vec{p}_c\right) \tag{2}$$

where,

$$J = \frac{1}{N} \sum\nolimits_{x_i \in X} \left(\vec{x}_i - \vec{p}_{y_i}\right)^T Q \left(\vec{x}_i - \vec{p}_{y_i}\right) \tag{3}$$

The goal is to derive the class-related prototypes $\{\vec{p}_c | c \in C^s\}$ to utilize in the distance function, and the parameter matrix Q. In the first sub-section, suitable prototypes are achieved with some theoretical proof. The second sub-section explains the proposed distance function as a case of the general form in Eq. (1) and provides the parameter tuning method. Finally, the kernelized distance learning is proposed with a large-margin approach.

3.1. Prototype Learning

In distance learning, it is assumed that prototypes have been provided. Therefore, as an initial phase, prototype learning should be performed. Inspired by Changpinyo et al. [25], the prototype of each class is initially set to the mean of associated instances. Then the mapping parameters are tuned to map the semantic vector of each class to its prototype and finally, the learned mapping function is used to obtain the prototypes for unseen classes. Since the parameters are tuned after prototype determination, as mentioned previously, this approach is called late learning. In this case, the parameters may not be successful to exactly generate the prototypes as a function of semantic vectors. However, in early learning, the parameters are adjusted in order to map the semantic vectors to such prototypes that are as near as possible to instances of the associated class label. In this case, prototypes are not specified before parameter learning, and the mapping functions directly produce them. Therefore, they may not be the best prototypes for the training instances. In the following, it is proved that, early and late learning both generate similar parameters if the mapping function is linear and the objective function is the mean square distance between instances and the prototypes of associated classes. Here, the general objective function of minimizing inner-class distance in Eq. (3) is reformulated in Eq. (4) for early learning.

$$J\left(W\right) = \frac{1}{N} \sum\nolimits_{x_i \in X} \left(\vec{x}_i - W\vec{a}_{y_i}\right)^T Q \left(\vec{x}_i - W\vec{a}_{y_i}\right) \tag{4}$$

The objective is reformed as shown in Eq. (5) where, X and A are the matrices of which i^{th} columns are $\vec{x}_i$ and $\vec{a}_{y_i}$, respectively.

$$J\left(W\right) = \frac{1}{N} tr\left(\left(X - WA\right)^T Q \left(X - WA\right)\right) \tag{5}$$

The matrix A is equal to A^sH where, $A^s = [\vec{a}_1, \ldots, \vec{a}_S]$ and i^{th} column of $H_{S\times N}$ is a one-hot vector e_c such that $y_i = c$. By equalizing the derivative of J respect to W to zero, the fact presented in Eq. (6) is derived.

$$\frac{\partial J}{\partial W} = 0 \Rightarrow QWAA^T = QXA^T \tag{6}$$

Note that matrix Q can be singular and A^T may have no right inverse and cannot be removed from the expressions.

Let $\bar{X} = [\mu_1, \ldots, \mu_S]$ and μ_c be the mean of the instances related to class label c. Sum of the instances of class c (can be represented by XH^T) does not change if each instance is replaced by the mean of instances. This statement is formulated in Eq. (7).

$$XH^T = \bar{X}HH^T \tag{7}$$

where, columns of $\bar{X}H$ represent the set of instances replaced with the mean of corresponding class (i.e., i^{th} column of $\bar{X}$ is μ_c if $y_i = c$). By using Eq. (7), Eq. (8) can be achieved.

$$QXA^T = QXH^TA^{sT} = Q\bar{X}HH^TA^{sT} = Q\bar{X}HA^T \tag{8}$$

Finally, from Eq. (6) and Eq. (8), Eq. (9) is derived.

$$QWAA^T = Q\bar{X}HA^T \tag{9}$$

In late learning, WA in Eq. (5) is replaced by the matrix of prototypes associated to each instance, P^sH where, $P^s = [\vec{p}_1, \ldots, \vec{p}_S]$. Hence, the objective is formulated as shown in Eq. (10).

$$J = \frac{1}{N} tr\left((X - P^sH)^T Q\,(X - P^sH)\right) \tag{10}$$

The matrix P^s is found by zeroing the derivative of J respect to P^s as shown in Eq. (11).

$$\frac{\partial J}{\partial P^s} = 0 \Rightarrow QP^sHH^T = QXH^T \tag{11}$$

Since HH^T is a diagonal matrix with the number of instances for each class on the diagonal, this matrix is invertible. Hence,

$$QP^s = QXH^T\left(HH^T\right)^{-1} = Q\bar{X}HH^T\left(HH^T\right)^{-1} = Q\bar{X} \tag{12}$$

In other words, $\vec{p}_c = \mu_c + N_Q\vec{v}$ where, N_Q is a matrix of which column space forms the null space of Q as $\mathbb{N}_Q$ (i.e., $\mathbb{N}_Q = \{N_Q\vec{v}$ for any free vector of coefficients $\vec{v}\}$). Specifically, each row of Q is orthogonal to each column of N_Q (i.e., $QN_Q = 0$). The free vector $\vec{v}$ is for extracting any point from the null space. After finding P^s, the optimum $\tilde{W}$ to minimize the Frobenius norm of errors $||P^sH - \tilde{W}A||_F$ leads to having Eq. (13).

$$\tilde{W}AA^T = P^sHA^T = \bar{X}HA^T \tag{13}$$

As demonstrated, $\tilde{W}$ in Eq. (13) can be used as W in Eq. (9). If Q is positive definite and if A is full row-rank (that means AA^T is invertible), uniquely, $W = \tilde{W} = \bar{X}HA^T\left(AA^T\right)^{-1}$. So, in case of using linear mapping function, early and late learning

both generate similar parameters if the objective function is the mean square distance between instances and their correct prototypes. This is why; the late learning is chosen in this paper and prototypes are selected as the mean of the instances from associated class labels shifted by the null space of Q (i.e., $\vec{p}_c = \mu_c + N_Q\vec{v}$). By inserting these prototypes in the defined distance and objective in Eq. (10), the null space can be ignored as shown in Eq. (14) and each prototype is simply represented by the mean of instances of the associated class. This simplification is achieved because, $QN_Q = 0$ and Q is symmetric.

$$J = tr\left((X - (\bar{X} + N_Q V)H)^T Q(X - (\bar{X} + N_Q V)H)\right) = tr\left(\left(X - \bar{X}H\right)^T Q\left(X - \bar{X}H\right)\right) \tag{14}$$

In order to decrease the effect of imbalanced property in ZSL datasets, $\tilde{W}$ is chosen regardless of the number of instances in each class. In this case, $\tilde{W}$ is not necessarily equal to W. The second modification is to use the non-linear form of Ridge regression called Kernel Ridge Regression. The linear model and its derivative respect to $\tilde{W}$ is presented in Eq. (15).

$$\begin{gathered} \min_{\tilde{W}} ||P^s - \tilde{W}A^s||_F^2 + \lambda||\tilde{W}||_F^2 \Rightarrow \\ \tilde{W}A^sA^{sT} + \lambda\tilde{W} = P^sA^{sT} \Rightarrow \tilde{W} = \frac{1}{\lambda}\left(P^s - \tilde{W}A^s\right)A^{sT} \end{gathered} \tag{15}$$

where, λ is the regularization parameter. With the above form, the variable $\tilde{W}'$ can be defined in Eq. (16) to reformulate Eq. (15) in Eq. (17).

$$\tilde{W}' = \frac{1}{\lambda}(P^s - \tilde{W}A^s) \tag{16}$$

$$\tilde{W} = \tilde{W}'A^{sT} \tag{17}$$

By reusing Eq. (17) in Eq. (16), the model of Eq. (18) is achieved:

$$\tilde{W}' = \frac{1}{\lambda}\left(P^s - \tilde{W}'A^{sT}A^s\right) \tag{18}$$

which implies:

$$\tilde{W}' = P^s\left(A^{sT}A^s + \lambda I_S\right)^{-1} \tag{19}$$

A kernel matrix $K^s_{S\times S} = A^{sT}A^s$ can be computed for the semantic vectors of seen classes as a pairwise similarity of the class vectors using a kernel function k, in Eq. (20).

$$\tilde{W}' = P^s(K^s + \lambda I_S)^{-1} \tag{20}$$

Then $\vec{p}^u_c$ as the visual prototype of the unseen class $c \in C^u$ can be derived by 21.

$$\vec{p}^u_c = \tilde{W}\vec{a}^u_c = \tilde{W}'A^{sT}\vec{a}^u_c = \tilde{W}'\vec{k}^u_c \tag{21}$$

where, $\vec{k}^u_c = A^{sT}\vec{a}^u_c$ is the pairwise similarity between the semantic vector of unseen class c and all seen classes.

3.2. Distance Learning

Considering the minimization of inner-class distance in Eq. (2) for the general distance function Eq. (1), the solution could be derived as $Q = \Sigma^{-1}$. However, another approach is followed in this paper with some constraints on Q. By decomposing Q to $M^T M$, the distance function can be presented as Eq. (22) and the objective function can be reformulated as Eq. (23).

$$d_v\left(\vec{x}_i, \vec{\mu}_{y_i}\right) = \left(\vec{x}_i - \vec{\mu}_{y_i}\right)^T M^T M \left(\vec{x}_i - \vec{\mu}_{y_i}\right) \tag{22}$$

$$J\left(M\right) = \frac{1}{N} \sum\nolimits_{x_i \in X} \left(M\vec{x}_i - M\vec{\mu}_{y_i}\right)^T \left(M\vec{x}_i - M\vec{\mu}_{y_i}\right) \tag{23}$$

In other words, each instance x is initially mapped to another point Mx. It leads to having the mean of class c equal to $M\mu_c$. Then, the simple Euclidean distance is applied. To decrease the complexity of finding the possible solutions in the diverse search space for M, it is assumed that the mapping function should not change the prototypes. In conclusion, each instance is initially mapped to $\mathbb{V}_p$ as the vector space of which the prototypes are the basis. The mapping matrix is idempotent (i.e., $MM = M$). In case of having symmetric M, then the mapping matrix is uniquely the perpendicular projection matrix on $\mathbb{V}_p$ (i.e., $M = M_{sym} = P^S\Big({P^S}^T P^S\Big)^{-1} {P^S}^T$). However, in case of having asymmetric M, this projection is not perpendicular. Anyway, $M\vec{\mu}_{y_i} = \vec{\mu}_{y_i}$, and the distance function can be presented as shown in Eq. (24).

$$d_v\left(\vec{x}_i, \vec{\mu}_c; M\right) = \left(M\vec{x}_i - \vec{\mu}_c\right)^T \left(M\vec{x}_i - \vec{\mu}_c\right) \tag{24}$$

Therefore, the objective function is remodeled as:

$$\begin{aligned} J\left(M\right) = \frac{1}{N} \sum\nolimits_{x_i \in X} \left(M\vec{x}_i - \vec{\mu}_{y_i}\right)^T \left(M\vec{x}_i - \vec{\mu}_{y_i}\right) = \frac{1}{N} ||MX - \bar{X}H||_F^2 \\ s.t.\ \forall \vec{\mu}_c : \quad M\vec{\mu}_c = \vec{\mu}_c \end{aligned} \tag{25}$$

However, these constraints can be ignored, because minimizing the objective function regardless of the constraints finds a solution for which the constraints are also held. To prove that, it is sufficient to show that the optimal M maps all instances to $\mathbb{V}_p$. Hence, M is a projection matrix on $\mathbb{V}_p$ and does not change any prototype. Assume that at least one instance $\vec{x}$ exists that is not mapped to $\mathbb{V}_p$ by M. The Euclidean distance between $M\vec{x}$ and the associated prototype $\vec{\mu}$ is addressed in Eq. 26.

$$||M\vec{x} - \vec{\mu}||^2 = ||\underbrace{\left(M\vec{x} - \vec{x}_p\right)}_{\overrightarrow{\Delta x}} + \underbrace{\left(\vec{x}_p - \vec{\mu}\right)}_{\overrightarrow{d_x}}||^2 = ||\overrightarrow{\Delta x}||^2 + ||\overrightarrow{d_x}||^2 + 2\underbrace{\overrightarrow{\Delta \mathrm{x}}^T \overrightarrow{d_x}}_{0} \tag{26}$$

where, $\vec{x}_p$ is equal to perpendicular projection of $M\vec{x}$ on $\mathbb{V}_p$ (i.e., $\vec{x}_p = M_{sym} M\vec{x}$). Hence, $\overrightarrow{d_x}$ as the difference vector of $\vec{\mu}$ and $\vec{x}_p$ is located on $\mathbb{V}_p$ and $\overrightarrow{\Delta x}$ as vector of difference between $M\vec{x}$ and $\vec{x}_p$ is in the complement space and is orthogonal respect to all the vectors in $\mathbb{V}_p$. This is why; the last term in Eq. (26) is equal to zero ($\overrightarrow{\Delta \mathrm{x}} \perp \overrightarrow{d_x}$). There is at least one instance for which, M does not map it to $\mathbb{V}_p$, and associated $\overrightarrow{\Delta x}$ is

not zero. In this case, replacing M by $M_{sym}M$ leads to have a smaller distance between the mapped instance and the prototype because $\overrightarrow{\Delta x}$ becomes zero. As a contrary, M is not the optimal solution unless it maps all the instances to $\mathbb{V}_p$ and as a consequence, all the constraints of Eq. (25) are held and can be removed from the model. By zeroing the derivative of J in Eq. (25) respect to M, the optimal matrix will be equal to $\bar{X}HX^T\left(XX^T\right)^{-1}$ if X is full-row rank.

3.3. Kernel-based Large-margin Discriminative Distance Learning

The weakness of the proposed model in Eq. (25) is that it focuses on minimizing inner-class distance without considering between-class discrimination which can be beneficial for ZSL. Therefore, in the following, as well as kernelizing the distance function in Eq. (24), a large-margin objective is proposed for learning the distance function in order to improve discriminative properties. At first, the distance function in Eq. (24) can be expanded and kernelized in Eq. (27) :

$$\begin{aligned} d_v\left(\vec{x}_i, \vec{\mu}_c; M\right) &= \left(M\vec{x}_i\right)^T\left(M\vec{x}_i\right) - 2\vec{\mu}_c^T M\vec{x}_i - \vec{\mu}_c^T\vec{\mu}_c \\ d_v^k\left(\vec{x}_i, \vec{\mu}_c; M\right) &= k\left(M\vec{x}_i,\ M\vec{x}_i\right) - 2k\left(M\vec{x}_i, \vec{\mu}_c\right) + k\left(\vec{\mu}_c,\ \vec{\mu}_c\right) \end{aligned} \tag{27}$$

where $k\left(.\right)$ is the applied kernel function. The goal is to learn M in order to minimize a loss function similar to the large-margin loss function used in [6] . However, that loss function was proposed based on the compatibility function in global compatibility learning. In this paper, that loss function is redesigned to support the distance functions for using in a space unifier model, e.g., prototype learning.

For each instance $\vec{x}_i$, it is desired to make the distance of the instance $\vec{x}_i$ to $\vec{\mu}_{y_i}$ at least m less than its distance to $\vec{\mu}_c$ which is the prototype of an arbitrary inconsistent class c. The hyper parameter m is the margin. Therefore, an error is considered if the margin m is violated. The loss imposed by the inconsistent class label c on the instance x_i is presented by Hinge loss as presented in Eq. 28.

$$h(\vec{x}_i, y_i, c; M) = \left[\mathrm{m} + d_v^k\left(\vec{x}_i, \vec{\mu}_{y_i}; M\right) - d_v^k\left(\vec{x}_i, \vec{\mu}_c; M\right)\right]_+ \tag{28}$$

where, the rectifier function is defined as $[v]_+ = \max\left(0, v\right)$.

The overall loss for the instance x_i and the final objective function for learning the proposed distance function are presented in Eq. (29) and Eq. (30), respectively.

$$\ell^{All}\left(x_i; M\right) = \sum_{c \in C^s - \{y_i\}} h(\vec{x}_i, y_i, c; M) \tag{29}$$

$$\min_M J_k = \sum_{\vec{x}_i \in X} \frac{1}{N_{y_i}} \ell^{All}\left(x_i; M\right) \tag{30}$$

where, N_{y_i} is the number of instances which belong to class y_i. The term $\frac{1}{N_{y_i}}$ is used to prevent the classes with many instances from influencing the objective function more than the others.

Stochastic Gradient Descent is used to optimize Eq. (30) and early stopping approach is used to prevent overfitting. The optimization algorithm is illustrated in Algorithm 1.

Table 2: Derivative of Hinge loss respect to M, using various kernels

Kernel name	Kernel formulation	Derivative of the Hinge loss respect to M
Linear	$k\left(\vec{a},\vec{b}\right) = \vec{a}^T\vec{b} + cnst$	$2\left(\vec{\mu}_c\vec{x}^T - \vec{\mu}_{c^*}\vec{x}^T\right)$
Quadratic	$k\left(\vec{a},\vec{b}\right) = \left(\vec{a}^T\vec{b} + cnst\right)^2$	$4\left[\left(cnst + \vec{x}^T M^T\vec{\mu}_c\right)\vec{\mu}_c - \left(cnst + \vec{x}^T M^T\vec{\mu}_{c^*}\right)\vec{\mu}_{c^*}\right]\vec{x}^T$
RBF	$k\left(\vec{a},\vec{b}\right) = \exp\left(-\gamma\|\vec{a}-\vec{b}\|^2\right)$	$4\gamma\left[\left(k\left(M\vec{x},\vec{\mu}_{c^*}\right) - k\left(M\vec{x},\vec{\mu}_c\right)\right)M\vec{x} + \left(k\left(M\vec{x},\vec{\mu}_c\right)\vec{\mu}_c - k\left(M\vec{x},\vec{\mu}_{c^*}\right)\vec{\mu}_{c^*}\right)\right]\vec{x}^T$

The parameter η is the learning rate of gradient descent, and the validation set is utilized to tune it. The value of this parameter is picked among the candidates regarding the results on the validation set. The gradient elements are computed and used in updating statement Eq. (31) for each selected instance $\vec{x}$ and the inconsistent class label c.

$$M^{new} = M^{old} - \frac{\eta}{N_{c^*}}\left(\frac{\partial\left(d_v^k\left(\vec{x},\vec{\mu}_{c^*};M\right)\right)}{\partial M} - \frac{\partial\left(d_v^k\left(\vec{x},\vec{\mu}_c;M\right)\right)}{\partial M}\right) \tag{31}$$

where, $c^* = y\left(x\right)$, for updating M. The derivative of $\mathbb{m} + d_v^k\left(\vec{x},\vec{\mu}_{c^*};M\right) - d_v^k\left(\vec{x},\vec{\mu}_c;M\right)$ varies concerning the utilized kernel function. The derivative formulations in case of using Linear kernel, Quadratic kernel and Polynomial kernel are illustrated in Table 2.

Algorithm 1 Optimization of large-margin distance function using Stochastic Gradient Descent

```
for T iterations do
    Draw a random instance (x⃗, c*) ∈ X
    for i = 1 : S − 1 do
        Draw a random label c ∈ C^s − {c*}
        if (m + d_v^k(x⃗, μ⃗_{c*}; M) − d_v^k(x⃗, μ⃗_c; M)) > 0 then
            Update M with gradient descent based on (31)
        end if
    end for
end for
```

3.4. Classification Task

During the test phase, the learned distance function is used to classify instances. In case of ZSL experiment, the class of $\vec{q}$, which belongs to one of unseen classes, is determined based on Eq. (32).

$$y\left(\vec{q}\right) = \min_{c\in C^u} d_v^k\left(\vec{q},\vec{p}_c^u;M\right) \tag{32}$$

On GZSL experiment that the instance may belong to either seen or unseen classes, the imbalance data issue will lead to predictions biasing toward seen classes [33]. To tackle this problem, a strategy similar to Calibrated Stacking (CS) [33] (decreasing the compatibility score of seen classes by a constant factor) is applied. In the proposed distance function, the weights of seen classes are reduced by adding a constant value α to the calculated distance from any seen class prototype. Therefore, the class of $\vec{q}$ is determined as:

$$y\left(\vec{q}\right) = \min_{c\in C^s\cup C^u} d_v^k\left(\vec{q},\vec{p}_c;M\right) + \alpha\mathbb{I}\left(c\in C^s\right) \tag{33}$$

Table 3: Statistics of the datasets used in the experiments

Dataset	Granularity	#Images	#Classes	#Attributes
aPY	course	15339	32	64
CUB	fine	11788	200	312
SUN	fine	14340	717	102
AWA1	course	30475	50	85
AWA2	course	37322	50	85

where $\mathbb{I}(c \in C^s) = 1$ if $c \in C^s$, otherwise, it is equal to 0. In the case of $c \in C^s$, $\vec{p}_c$ is calculated as the mean of the seen class instances $\vec{\mu}_c$ and for $c \in C^u$, $\vec{p}_c$ is estimated by Eq. (21).

4. Experiments and Evaluation

The proposed method is evaluated on five ZSL benchmark datasets: Attribute Pascal and Yahoo (aPY) [2], Caltech-UCSD-Birds (CUB) [34], SUN Attributes (SUN) [35], Animals with attributes (AWA1) [36], and Animals with Attributes 2 (AWA2) [37]. The statistics of the utilized datasets are illustrated in Table 3. The top-layer pooling units of Resnet-101 [38], with 2048-dimensions, pre-trained on ImageNet dataset, are used as the visual features of images provided by [37]. In order to have a fair comparison, provided attributes with each class are used as the semantic class embeddings. Also, in the experiments, the data splitting proposed in [37, 39] (called proposed split) shown in Table 4 is applied that guarantees uncommon categories between unseen classes and ImageNet classes which the visual model is pre-trained on.

For distance learning, the initial value of M is set to the identity matrix I_{D_v}. The learning rate η and regularization parameter λ are selected by cross-validation technique and by using the validation splits of seen classes provided with the proposed splits in [37, 39]. Also, in Kernel Ridge Regression, Quadratic kernel is utilized to learn the prototypes of unseen classes and the constant bias factor α is set to 0.03 in GZSL for all datasets. This value is a general value used for all datasets and is extracted experimentally.

Average per-class top-1 accuracy is used for evaluating methods in ZSL setting, which is calculated as presented in Eq. (34).

$$Acc^U = \frac{1}{||C^u||} \sum_{c \in C^u} \frac{\#\ of\ correct\ predictions\ in\ class\ c}{\#\ of\ instances\ in\ class\ c} \tag{34}$$

where, $||C^u||$ is the number of classes in C^u. This metric averages the correct predictions in each class independently to prevent the populated classes make an extra impact on the final measurement. For GZSL setting, besides reporting distinct average per-class accuracies of seen classes (Acc^S) and unseen classes (Acc^U), the Harmonic Mean (H) of these measures are also reported as a unified metric for evaluating the performance of

Table 4: Datasets split used in the experiments proposed by [37]

Dataset	#Seen/Unseen Classes	#images of seen classes #images in GZSL training/testing	#images of unseen classes
aPY	20/12	7415 5932/1483	7924
CUB	150/50	8821 7057/1764	2967
SUN	645/72	12900 10320/2580	1440
AWA1	40/10	24790 19832/4958	5685
AWA2	40/10	29409 23527/5882	7913

the methods. This metric is calculated as presented in Eq. (35).

$$H = \frac{2 \times Acc^U \times Acc^S}{Acc^U + Acc^S} \tag{35}$$

For both ZSL and GZSL, the average ranking among all methods is also reported, which is calculated by averaging the performance ranking of the method on all the datasets.

4.1. Kernel Effect on Distance Learning

To investigate the impact of kernel function on distance learning, the proposed model is trained with various kernels on CUB dataset. Linear kernel, Quadratic kernel and RBF kernel are considered for this experiment. The results are presented in Fig. 1. The RBF kernel achieves the best result with an average accuracy of 59.1% followed by Linear and Quadratic kernels with 50% and 49%, respectively. The variation in the results by using different kernels emphasizes the effect of kernel employment in distance function learning that can lead to an improvement in the case of proper choice. For all the following experiments, RBF kernel is utilized with the value of its free parameter γ set to 1/2048 (the number of features in the visual feature space). The margin $\mathbb{m}$ in Eq. (28) is also set to 0.05 in all experiments.

4.2. Comparison with State-of-the-art Methods

From the viewpoint of effectiveness, the proposed method is compared with 20 state-of-the-art methods: DAP (Lampert et al. [36]), CONSE (Norouzi et al. [40]), SSE (Zhang and Saligrama [13]), LATEM (Xian et al. [41]), ALE (Akata et al. [7]), DEVISE (Frome et al. [6]),SJE (Akata et al. [29]), ESZSL (Romera-Paredes and Torr [42]), SYNC (Changpinyo et al. [43]), SAE (Kodirov et al. [8]), GFZSL (Verm and Rai [10]), PSRZSL (Annadani and Biswas [44]), DEM (Zhang et al. [11]), TVN (Zhang et al. [45]), RNet (Sung et al. [30]), AML (Jiang et al. [5]), EXEM (Changpinyo et al. [46]), MLSE (Ding and Liu [47]), AUVS (Zhang et al. [48]) and PLNPS (Zhang et al. [4]).

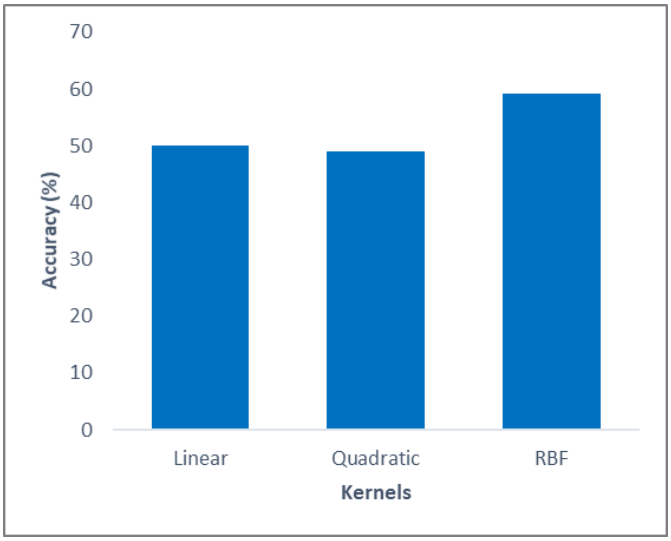

Figure 1: The impact of using kernel in distance function learning on CUB dataset for zero-shot recognition.

4.2.1. Zero-Shot Learning

The comparison results on conventional ZSL based on average per-class accuracy are demonstrated in Table 5. According to the results, the proposed method achieves the best performance on three out of five datasets and the second-ranked in the other two datasets. On AWA1 and AWA2, the proposed method surpasses the second-ranked methods TVN and MLSE with 3.6% and 2.9%, respectively. On SUN dataset, which contains fewer per-class instances and more classes compared to other datasets, the proposed method also performs well by achieving 63.6% that is at least 0.7% greater than the other methods. On aPY, in which weak relations exist between the seen and unseen classes, the proposed method obtains the second place (after MLSE) with an average of 43.3%. On CUB, a fine-grained dataset with highly similar classes, MLSE also attains the best result with an average per-class accuracy of 64.2%, while the proposed method scores 59.1% as the second rank. That may be caused by the less discriminative properties of classes in visual feature space. Finally, by considering the average ranking, the proposed method confirms its overall advantage by achieving the average ranking of 1.4, which is 0.4 lower than the second-best method, MLSE. Despite the simplicity of the proposed approach, it achieves remarkable results compared to state-of-the-art ZSL methods with various approaches. It should be notified that MLSE could not be successful in GZSL evaluations as followed.

4.2.2. Generalized Zero-Shot Learning

In realistic scenarios, the assumption that all the test instances just belong to unseen classes is unreasonable. Therefore, an experiment with GZSL setting is also considered in which the model has to predict both seen and unseen instances during the test phase. The performances of different approaches in GZSL are compared in Table 6. In this report, two strategies have been considered. In one of them, seen and unseen classes are taken equally into consideration (setting $\alpha = 0$ in Eq. (33)), and in the other one, the

Table 5: Zero-shot recognition in terms of average per-class accuracy (%) on five benchmark datasets. In the case of no available reported result, '-' is used. The best result is made bold and the second-best is underlined.

Method	aPY	CUB	SUN	AWA1	AWA2	Avg Rank
DAP [36]	33.8	40.0	39.9	44.1	46.1	17.6
CONSE [40]	26.9	34.3	38.8	45.6	44.5	18.4
SSE [13]	34.0	43.9	51.5	60.1	61.0	14
LATEM [41]	35.2	49.3	55.3	55.1	55.8	13.9
ALE [7]	39.7	54.9	58.1	59.9	62.5	9.5
DEVISE [6]	39.8	52.0	56.5	54.2	59.7	11.8
SJE [29]	32.9	53.9	53.7	65.6	61.9	12.1
ESZSL [42]	38.3	53.9	54.5	58.2	58.6	12.3
SYNC [43]	23.9	55.6	56.3	54.0	46.6	13.9
SAE [8]	8.3	33.3	40.3	53.0	54.1	17.8
GFZSL [10]	38.4	49.3	60.6	68.3	63.8	8.9
PSRZSL [44]	38.4	56.0	61.4	-	63.8	7
DEM [11]	35.0	51.7	61.9	68.4	67.1	7.6
TVN [45]	41.3	58.1	60.7	<u>68.8</u>	-	4.3
RNet [30]	-	55.6	-	68.2	64.2	5.8
AML [5]	41.6	57.5	58.1	65.3	-	7.4
EXEM [46]	-	58	<u>62.9</u>	68.1	64.6	4
MLSE [47]	**46.2**	**64.2**	62.8	-	<u>67.8</u>	<u>1.8</u>
AUVS [48]	40.1	52.6	61.7	67.4	-	7.9
PLNPS [4]	42.8	53.2	60.4	67.4	-	7.9
Proposed Method	<u>43.3</u>	<u>59.1</u>	**63.6**	**72.4**	**70.7**	**1.4**

weighted method is used to reduce the impact of seen classes (setting $\alpha = 0.03$ in Eq. (33)). The value of α in the weighting method is achieved experimentally as a good value for all datasets.

According to Table 6, the proposed method outperforms the existing approaches in GZSL problem similar to conventional ZSL experiment. Achieving a superior performance by both versions compared to the other methods, regarding harmonic mean, demonstrates the better generalization ability of our method. Although the simple form of the proposed method is biased towards seen classes to some extent, its overal good performance makes it possible to achieve a considerable balance by the weighting method. As it can be seen, weighing strategy increases harmonic mean by making a more balanced performance among seen and unseen classes that demonstrates the high potential of the proposed approach.

In terms of the harmonic mean, the proposed weighted method outperforms the competitors on four out of five datasets by a significant superiority. On CUB, the method achieves the highest performance jointly with RNet. This remarkable performance results in obtaining the best average ranking while the simple proposed approach attains the second-best place. In terms of average per-class accuracy on unseen classes, the weighted proposed method surpasses the others on all five datasets, while the simple approach stands on the second place on aPY, AWA1, and AWA2. On the datasets CUB and SUN, RNet and AUVS attain the second top ranks, respectively. Regarding the accuracy on

Table 6: Generalized zero-shot recognition on aPY, CUB, SUN, AWA1 and AWA2. Acc^U = average per-class accuracy of unseen classes, Acc^S = average per-class accuracy of seen classes and H = harmonic mean. In case of no available reported results, '-' is used. The best result is made bold and the second best is underlined.

Method	aPY			CUB			SUN			AWA1			AWA2			H Avg
	Acc^U	Acc^S	H	Acc^U	Acc^S	H	Acc^U	Acc^S	H	Acc^U	Acc^S	H	Acc^U	Acc^S	H	Rank
DAP [36]	4.8	78.3	9	1.7	67.9	3.3	4.2	25.1	7.2	0	88.7	0	0	84.7	0	18
CONSE [40]	0	**91.2**	0	1.6	**72.2**	3.1	6.8	39.9	11.6	0.4	88.6	0.8	0.5	90.6	1	19.1
SSE [13]	0.2	78.9	0.4	8.5	46.9	14.4	2.1	36.4	4	7	80.5	12.9	8.1	82.5	14.8	16.6
LATEM [41]	0.1	73	0.2	15.2	57.3	24	14.7	28.8	19.5	7.3	71.7	13.3	11.5	77.3	20	14.4
ALE [7]	4.6	73.7	8.7	23.7	62.8	34.4	21.8	33.1	26.3	16.8	76.1	27.5	14	81.8	23.9	9.9
DEVISE [6]	4.9	76.9	9.2	23.8	53	32.8	16.9	27.4	20.9	13.4	68.7	22.4	17.1	74.7	27.8	11
SJE [29]	3.7	55.7	6.9	23.5	59.2	33.6	14.7	30.5	19.8	11.3	74.6	19.6	8	73.9	14.4	13
ESZSL [42]	2.4	70.1	4.6	12.6	63.8	21	11	27.9	15.8	6.6	75.6	12.1	5.9	77.8	11	15.4
SYNC [43]	7.4	66.3	13.3	11.5	70.9	19.8	7.9	43.3	13.4	8.9	87.3	16.2	10	90.5	18	13.6
SAE [8]	0.4	80.9	0.9	7.8	54	13.6	8.8	18	11.8	1.8	77.1	3.5	1.1	82.2	2.2	17.3
GFZSL [10]	0	83.3	0	0	45.7	0	0	39.6	0	1.8	80.3	3.5	2.5	80.1	4.8	19.2
PSRZSL [44]	13.5	51.4	21.4	24.6	54.3	33.9	20.8	37.2	26.7	-	-	-	20.7	73.8	32.3	8.3
DEM [11]	11.1	75.1	19.4	19.6	57.9	29.2	20.5	34.3	25.6	32.8	84.7	47.3	30.5	86.4	45.1	8.6
TVN [45]	16.1	66.9	25.9	26.5	62.3	37.2	22.2	38.3	28.1	27	67.9	38.6	-	-	-	6
RNet [30]	-	-	-	38.1	61.4	**47**	-	-	-	31.4	**91.3**	46.7	30	**93.4**	45.3	3.8
AML [5]	12.6	74.5	21.5	25.7	66.6	37.1	20	38.2	26.3	11.8	89.6	20.8	-	-	-	8.4
EXEM [46]	-	-	-	28	67.8	39.6	14.6	42	21.6	31.6	88.1	46.5	30.8	89.3	45.8	6.3
MLSE [47]	12.7	74.3	21.7	22.3	71.6	34	20.7	36.4	26.4	-	-	-	23.8	83.2	37	7.5
AUVS [48]	27.5	70.6	39.6	31.5	40.2	35.3	41.2	26.7	32.4	38.7	74.6	51	-	-	-	4.8
PLNPS [4]	25.9	79.5	39.1	37.8	58.2	45.9	39.7	38.9	39.3	37	84.7	51.4	25.9	79.5	39.1	3.6
Proposed-Simple	31.2	79.5	44.8	28.4	63	39.2	27.9	**43.4**	34	45.9	85.6	59.8	45	88.1	59.6	2.8
Proposed-Weighted	**37.8**	68.8	**48.8**	**49.7**	44.6	**47**	**48.6**	35.1	**40.8**	**59.2**	79.5	**67.9**	**57.7**	83.2	**68.1**	**1.1**

seen classes, the best performance belongs to CONSE on aPY and CUB. On SUN, our unweighted approach achieves the highest accuracy, and RNet scores the best on AWA1 and AWA2.

According to the results, some methods perform poorly on unseen classes despite the high accuracy on seen classes due to the overfitting problem. The overall superiority of the proposed approaches shows that it has kept the performance on seen classes high as well as improving the accuracy of unseen classes.

4.2.3. Comparison with GAN-based Methods

The proposed method is not comparable to GAN-based approaches from the perspective of methodology, architecture, and complexity. However, due to the popularity that GANs for ZSL and specially GZSL have gained in recent years, the proposed method is compared with 6 state-of-the-art GAN-based approaches that have been proposed recently. These methods are f-CLSWGAN (Xian et al. [17]), Cycle-CLSWGAN (Felix et al. [18]), LisGAN (Li et. al [19]), GDAN (Huang et al. [49]), DASCN (Ni et al. [50]) and LsrGAN (Vyas et al. [16]). Among these models, GDAN and DASCN have not reported any result for ZSL. Thus, those are considered just for GZSL evaluation.

The comparison on ZSL is shown in Table 7. As None of the methods have evaluated their performance on AWA2 for ZSL task, this dataset has been ignored here. It can be seen that the proposed method performs better also in comparison with GAN-based approaches. Obtaining the highest accuracy on aPY, SUN, and AWA1 and scoring the second-best performance on CUB leads to achieving the lowest average ranking with a superiority of 1 compared to LsrGAN as the rival method. This method attains the best accuracy on CUB.

The GZSL results on the five datasets are reported in Table 8. Despite the simplicity

Table 7: Comparison with GAN-based methods on zero-shot recognition task evaluated on aPY, CUB, SUN, and AWA1. In case of no available reported results, '-' is used. The best result is made bold and the second best is underlined.

Method	aPY	CUB	SUN	AWA1	Avg Rank
f-CLSWGAN [17]	40.5	57.3	60.8	68.2	3.8
Cycle-CLSWGAN [18]	-	58.4	60	66.3	4.7
LisGan [19]	<u>43.1</u>	58.8	61.7	<u>70.6</u>	2.5
LsrGAN [16]	-	**60.3**	<u>62.5</u>	66.4	<u>2.3</u>
Proposed Method	**43.3**	<u>59.1</u>	**63.6**	**72.4**	**1.3**

Table 8: Comparison with GAN-based methods on Generalized zero-shot recognition task evaluated on aPY, CUB, SUN, AWA1 and AWA2. Acc^U = average per-class accuracy of unseen classes, Acc^S = average per-class accuracy of seen classes and H = harmonic mean. In case of no available reported results, '-' is used. The best result is made bold and the second best is underlined.

Method	aPY			CUB			SUN			AWA1			AWA2			H Avg
	Acc^U	Acc^S	H	Acc^U	Acc^S	H	Acc^U	Acc^S	H	Acc^U	Acc^S	H	Acc^U	Acc^S	H	Rank
f-CLSWGAN [17]	32.9	61.7	42.9	43.7	57.7	49.7	42.6	36.6	39.4	57.9	61.4	59.6	-	-	-	5.8
Cycle-CLSWGAN [18]	-	-	-	45.7	<u>61</u>	<u>52.3</u>	33.6	<u>49.4</u>	40	56.9	64	60.2	-	-	-	4.3
LisGan [19]	34.3	68.2	45.7	46.5	57.9	51.6	42.9	37.8	40.2	52.6	<u>76.3</u>	62.3	-	-	-	3.9
GDAN [49]	30.4	**75**	43.4	39.3	**66.7**	49.5	38.1	**89.9**	**53.4**	-	-	-	<u>32.1</u>	<u>67.5</u>	<u>43.5</u>	3.3
DASCN [50]	**39.7**	59.5	<u>47.6</u>	45.9	59	51.6	42.4	38.5	40.3	59.3	68	<u>63.4</u>	-	-	-	2.9
LsrGAN [16]	-	-	-	<u>48.1</u>	59.1	**53**	<u>44.8</u>	37.7	<u>40.9</u>	54.6	74.6	63	-	-	-	**2**
Proposed-Weighted	<u>37.8</u>	<u>68.8</u>	**48.8**	**49.7**	44.6	47	**48.6**	35.1	40.8	<u>59.2</u>	**79.5**	**67.9**	**57.7**	**83.2**	**68.1**	<u>2.6</u>

and less complexity of the proposed method, its results are comparable to GAN-based approaches that mainly focus on GZSL task. According to the results, the best performance on aPY, AWA1, and AWA2, in terms of harmonic mean, belongs to the proposed method which leads to obtaining the second best harmonic mean average ranking, while LsrGAN obtains the best average ranking. Regarding the average per-class accuracy of unseen classes, the proposed method attains the highest accuracy on CUB, SUN, and AWA2, while the best performance on aPY and AWA1 belongs to DASCN. Moreover, in terms of average per-class accuracy of seen classes, the highest accuracy is achieved by GDAN on aPY, CUB and SUN. Our approach scores the best on AWA1 and AWA2.

Although most of the methods have not been evaluated on AWA2, similar properties with AWA1 result in expecting a trend like the one on AWA1 in which our method performed better than the competitors.

4.3. Evaluating Distance Learning

In order to investigate the effect of learned distance function, the performance of the proposed method is compared with a base method in which the value of M in distance function is set to identity matrix I_{D_v}. The comparison results on zero-shot and Generalized zero-shot recognition are shown in Figs. 2 and 3, respectively. For the GZSL task, the simple version which does not reduce the weight of seen classes is considered. It can be seen that the learned distance function improves the model in zero-shot recognition on all five datasets. This trend also occurs in generalized zero-shot recognition where the learned distance function causes a remarkable improvement of harmonic mean by raising the average per-class accuracy of unseen classes as well as keeping the average per-class accuracy of seen classes high on all five considered datasets.

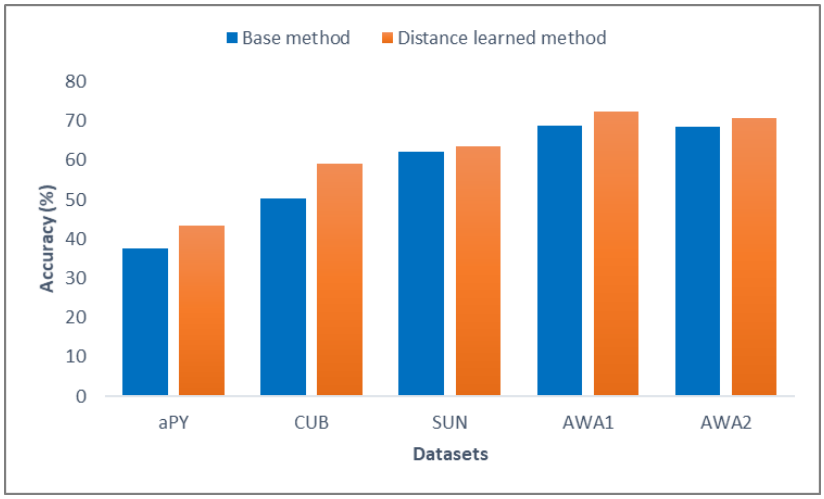

Figure 2: Study of the distance function learning in case of zero-shot recognition. The distance learning is eliminated in the base method by setting M to identity matrix I_{D_v} in the distance function.

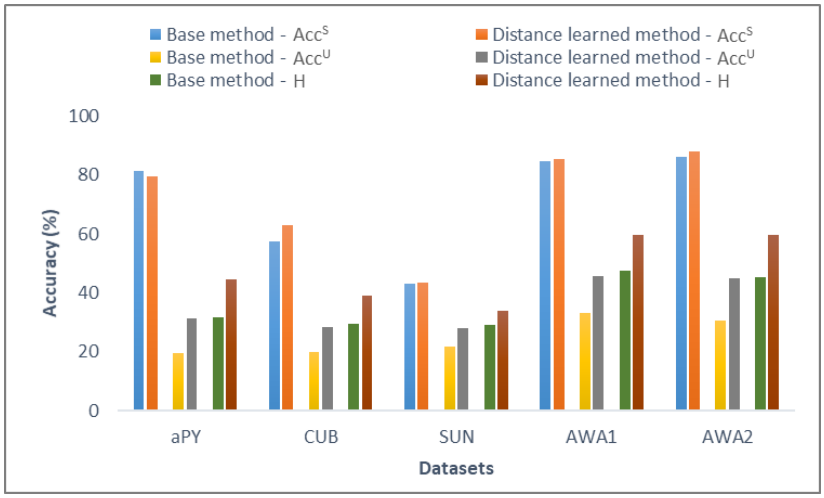

Figure 3: Study of the distance function learning in case of generalized zero-shot recognition. The distance learning is eliminated in the base method by setting M to Identity matrix I_{D_v} in the distance function.

4.4. Parameters Influence

The influence of RBF kernel free parameter γ and the margin $\mathbb{m}$ in the final performance is analyzed by taking CUB as an example and training the proposed model with various settings. The results are illustrated in Figure 4. In case of using RBF kernel, the hinge loss used in Eq. (28) for an arbitrary instance $\vec{x}_i$ with the inconsistent class label c is formulated as in Eq. (36).

$$\left[2\exp\left(-\gamma||M\vec{x}_i - \vec{p}_c||^2\right) - 2\exp\left(-\gamma||M\vec{x}_i - \vec{p}_{y_i}||^2\right) + \mathbb{m}\right]_+ \tag{36}$$

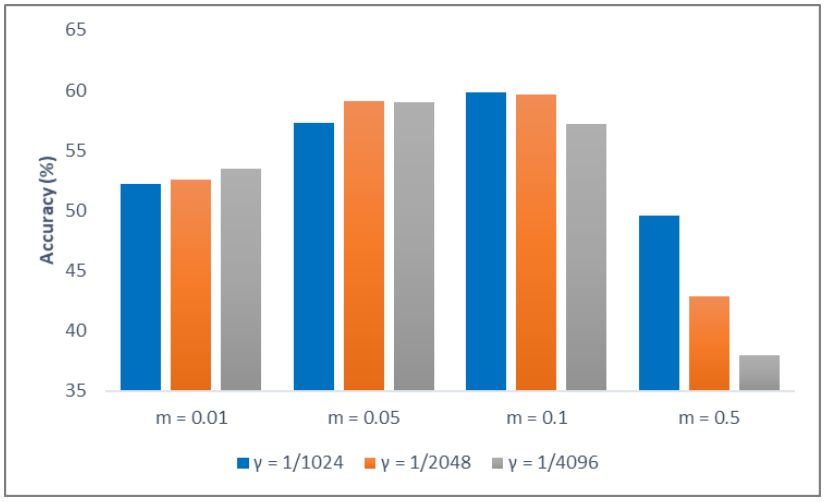

Figure 4: The influence of RBF kernel free parameter γ and the margin m on average per class accuracy on CUB dataset.

As it is clear, the internal expression value is bounded with the range $[-2+\mathrm{m}, 2+\mathrm{m}]$. Thus, choosing a large m results in an ineffective training procedure. Regarding this point, m has been chosen from $\{0.01,\ 0.05,\ 0.1, 0.5\}$. Furthermore, with respect to the number of features in visual feature space, which is 2048, the numbers $\left\{\frac{1}{1024}, \frac{1}{2048}, \frac{1}{4096}\right\}$ have been considered for the value of γ to construct several settings for the experiments.

According to Fig. 4, the setting with $(\gamma = \frac{1}{1024}, \mathrm{m} = 0.1)$ obtains the highest accuracy among all settings while the least performance belongs to $(\gamma = \frac{1}{4096}, \mathrm{m} = 0.5)$. It is observed that choosing a very small margin $(\mathrm{m} = 0.01)$ in all experiments with various values of γ results in a degraded performance compared to utilizing higher values of margin $(\mathrm{m} = 0.05$ and $\mathrm{m} = 0.1)$. However, increasing the value of margin remarkably also reduces the accuracy as it can be seen that setting m to 0.5 results in a significant decline in the performance. Although the fixed setting $(\gamma = \frac{1}{2048}, \mathrm{m} = 0.05)$ is used for the other experiments on all datasets in this paper, achieving higher performance with some other parameter settings shows that utilizing some validation procedures such as cross-validation technique for choosing the best setting may enhance the performance of the proposed distance function, more than the reported results.

4.5. Visual Prototypes Study

As discussed in Section 3, the proposed distance function does not affect the prototypes. Thus, choosing appropriate prototypes with discriminative properties, to some degree, is essential. To illustrate this property for the visual prototypes as mean of the instances, the instances and their means for seen classes are visualized with t-SNE on AWA2 in Fig. 5. Although there are numerous instances that affect the discriminative property negatively by crossing their approximate class margins, the means of the classes possess a proper distance to each other, making them suitable prototypes for the proposed distance function.

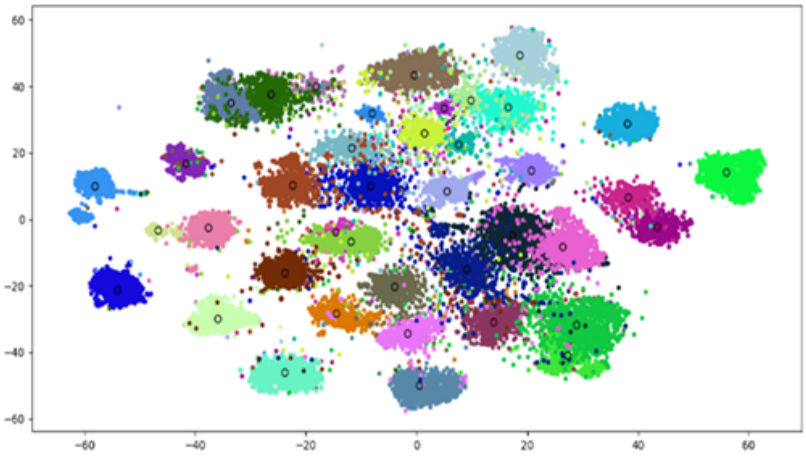

Figure 5: t-SNE visualization of instances and their means as class prototypes on AWA2 seen classes. The prototypes are illustrated with a black circle.

4.6. Kernel Effect on Learnt Visual Prototypes

The prototypes of unseen classes are approximated using Kernel Ridge Regression learned on seen prototypes as described in Section 3. In this subsection, the influence of using different kernels for Kernel Ridge Regression is examined on the final performance of ZSL. The results on AWA2 as an example are illustrated in Figure 6. The difference between the performance of the Quadratic kernel as the best performer and the RBF kernel that obtains the second-best is just 0.1%. The lowest accuracy belongs to the Linear kernel with a 1% difference from the Quadratic kernel. It is observed that the choice of the kernel for prototype learning is not as significant as choosing the proper kernel for distance function learning based on the difference in performance of considered kernels.

5. Conclusion

In this paper, a non-linear large-margin distance function is proposed for Zero-Shot Learning, aiming to tune the distance between an instance and the class-related prototypes. Moreover, a mapping function is learned to construct the visual class prototypes of unseen classes from semantic information to bring the possibility of utilizing the learned distance function for zero-shot and generalized zero-shot recognition tasks. Conducted experiments on five ZSL benchmark datasets demonstrated that the proposed approach outperforms state-of-the-art methods in most cases under ZSL and generalized ZSL settings. Considering other metrics in kernel-based distance/similarity learning such as Cosine, extending the proposed theoretical proof to support other objective functions (e.g., Hinge loss), investigating the effect of noise on the proposed approach and extending the GAN based architectures by the proposed objective function are some tasks that may be considered in the future.

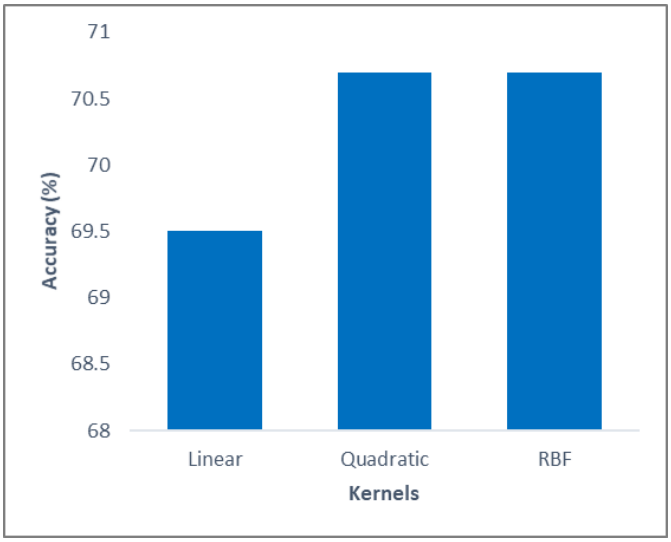

Figure 6: The performance of the model in zero-shot recognition on AWA2 using different kernels for prototype learning.

References

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

[2] A. Farhadi, I. Endres, D. Hoiem, D. Forsyth, Describing objects by their attributes, in: 2009 IEEE Conference on Computer Vision and Pattern Recognition, Institute of Electrical and Electronics Engineers (IEEE), 2009, pp. 1778–1785. doi:10.1109/cvpr.2009.5206772.

[3] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, J. Dean, Distributed Representations of Words and Phrases and their Compositionality, in: NIPS'13 Proceedings of the 26th International Conference on Neural Information Processing Systems, 2013, pp. 3111–3119. arXiv:1310.4546.

[4] H. Zhang, H. Mao, Y. Long, W. Yang, L. Shao, A Probabilistic Zero-Shot Learning Method via Latent Nonnegative Prototype Synthesis of Unseen Classes, IEEE Transactions on Neural Networks and Learning Systems 31 (7) (2020) 2361–2375. doi:10.1109/TNNLS.2019.2955157.

[5] H. Jiang, R. Wang, S. Shan, X. Chen, Adaptive metric learning for zero-shot recognition, IEEE Signal Processing Letters 26 (9) (2019) 1270–1274. doi:10.1109/LSP.2019.2917148.

[6] A. Frome, G. S. Corrado, J. Shlens, S. Bengio, J. Dean, M. Ranzato, T. Mikolov, DeViSE: A Deep Visual-Semantic Embedding Model, in: NIPS'13 Proceedings of the 26th International Conference on Neural Information Processing Systems, 2013, pp. 2121–2129.

[7] Z. Akata, F. Perronnin, Z. Harchaoui, C. Schmid, Label-Embedding for Image Classification, IEEE Transactions on Pattern Analysis and Machine Intelligence 38 (7) (2016) 1425–1438. arXiv:1503.08677, doi:10.1109/TPAMI.2015.2487986.

[8] E. Kodirov, T. Xiang, S. Gong, Semantic autoencoder for zero-shot learning, in: Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Vol. 2017-January, Institute of Electrical and Electronics Engineers Inc., 2017, pp. 4447–4456. arXiv:1704.08345, doi:10.1109/CVPR.2017.473.

[9] Z. Ji, Y. Yu, Y. Pang, J. Guo, Z. Zhang, Manifold regularized cross-modal embedding for zero-shot learning, Information Sciences 378 (2017) 48–58. doi:https://doi.org/10.1016/j.ins.2016.10.025.

[10] V. K. Verma, P. Rai, A Simple Exponential Family Framework for Zero-Shot Learning, in: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Vol. 10535 LNAI, Springer Verlag, 2017, pp. 792–808. arXiv:1707.08040, doi:10.1007/978-3-319-71246-8_48.

[11] L. Zhang, T. Xiang, S. Gong, Learning a Deep Embedding Model for Zero-Shot Learning, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 2021–2030.

[12] C. Pan, J. Huang, J. Hao, J. Gong, Towards zero-shot learning generalization via a cosine distance loss, Neurocomputing 381 (2020) 167–176. doi:10.1016/j.neucom.2019.11.011.

[13] Z. Zhang, V. Saligrama, Zero-shot learning via semantic similarity embedding, in: Proceedings of the IEEE International Conference on Computer Vision, 2015. doi:10.1109/ICCV.2015.474.

[14] H. Liu, L. Yao, Q. Zheng, M. Luo, H. Zhao, Y. Lyu, Dual-stream generative adversarial networks for distributionally robust zero-shot learning, Information Sciences 519 (2020) 407–422. doi:https://doi.org/10.1016/j.ins.2020.01.025.

[15] J. Liu, Z. Zhang, G. Yang, Cross-class generative network for zero-shot learning, Information Sciences 555 (2021) 147–163. doi:https://doi.org/10.1016/j.ins.2020.12.063.

[16] M. R. Vyas, H. Venkateswara, S. Panchanathan, Leveraging Seen and Unseen Semantic Relationships for Generative Zero-Shot Learning, in: European Conference on Computer Vision, Vol. 12375 LNCS, Springer Science and Business Media Deutschland GmbH, 2020, pp. 70–86. arXiv:2007.09549, doi:10.1007/978-3-030-58577-8_5.

[17] Y. Xian, T. Lorenz, B. Schiele, Z. Akata, Feature Generating Networks for Zero-Shot Learning, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2018, pp. 5542–5551. arXiv:1712.00981, doi:10.1109/CVPR.2018.00581.

[18] R. Felix, V. Kumar B G, I. Reid, G. Carneiro, Multi-modal Cycle-consistent Generalized Zero-Shot Learning, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 21–37.

[19] J. Li, M. Jing, K. Lu, Z. Ding, L. Zhu, Z. Huang, Leveraging the invariant side of generative zero-shot learning, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 7402–7411. arXiv:1904.04092, doi:10.1109/CVPR.2019.00758.

[20] Y. Shigeto, I. Suzuki, K. Hara, M. Shimbo, Y. Matsumoto, Ridge regression, hubness, and zero-shot learning, in: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Vol. 9284, Springer Verlag, 2015, pp. 135–151. arXiv:1507.00825, doi:10.1007/978-3-319-23528-8_9.

[21] Z. Ji, J. Wang, Y. Yu, Y. Pang, J. Han, Class-specific synthesized dictionary model for Zero-Shot Learning, Neurocomputing 329 (2019) 339–347. doi:10.1016/j.neucom.2018.10.069.

[22] M. Bucher, S. Herbin, F. Jurie, Improving semantic embedding consistency by metric learning for zero-shot classiffication, in: European Conference on Computer Vision, ECCV 2016, Vol. 9909 LNCS, Springer Verlag, 2016, pp. 730–746. arXiv:1607.08085, doi:10.1007/978-3-319-46454-1_44.

[23] Y. Wang, H. Zhang, Z. Zhang, Y. Long, L. Shao, Learning discriminative domain-invariant prototypes for generalized zero shot learning, Knowledge-Based Systems 196 (2020) 105796. doi:https://doi.org/10.1016/j.knosys.2020.105796.

[24] Y. Fu, T. M. Hospedales, T. Xiang, S. Gong, Transductive Multi-View Zero-Shot Learning, IEEE Transactions on Pattern Analysis and Machine Intelligence 37 (11) (2015) 2332–2345. arXiv:1501.04560, doi:10.1109/TPAMI.2015.2408354.

[25] S. Changpinyo, W. L. Chao, F. Sha, Predicting Visual Exemplars of Unseen Classes for Zero-Shot Learning, in: Proceedings of the IEEE International Conference on Computer Vision, Vol. 2017-October, Institute of Electrical and Electronics Engineers Inc., 2017, pp. 3496–3505. arXiv:1605.08151, doi:10.1109/ICCV.2017.376.

[26] H. Jiang, R. Wang, S. Shan, X. Chen, Learning class prototypes via structure alignment for zero-shot recognition, in: European Conference on Computer Vision, ECCV 2018, Vol. 11214 LNCS, Springer Verlag, 2018, pp. 121–138. arXiv:1807.09123, doi:10.1007/978-3-030-01249-6_8.

[27] T. Mensink, J. Verbeek, F. Perronnin, G. Csurka, Distance-based image classification: Generalizing to new classes at near-zero cost, IEEE Transactions on Pattern Analysis and Machine Intelligence 35 (11) (2013) 2624–2637. doi:10.1109/TPAMI.2013.83.

[28] T. Joachims, Optimizing search engines using clickthrough data, in: Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Association for Computing Machinery (ACM), New York, New York, USA, 2002, pp. 133–142. doi:10.1145/775047.775067.

[29] Z. Akata, S. Reed, D. Walter, H. Lee, B. Schiele, Evaluation of output embeddings for fine-grained image classification, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Vol. 07-12-June, IEEE Computer Society, 2015, pp. 2927–2936. arXiv:1409.8403, doi:10.1109/CVPR.2015.7298911.

[30] F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H. Torr, T. M. Hospedales, Learning to Compare: Relation Network for Few-Shot Learning, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2018, pp. 1199–1208. arXiv:1711.06025, doi:10.1109/CVPR.2018.00131.

[31] H. Zhang, Y. Long, W. Yang, L. Shao, Dual-verification network for zero-shot learning, Information Sciences 470 (2019) 43–57. doi:https://doi.org/10.1016/j.ins.2018.08.048.

[32] H. Zhang, P. Koniusz, Zero-shot kernel learning, in: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018, pp. 7670–7679. doi:10.1109/CVPR.2018.00800.

[33] W. L. Chao, S. Changpinyo, B. Gong, F. Sha, An empirical study and analysis of generalized zero-shot learning for object recognition in the wild, in: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), Vol. 9906 LNCS, Springer Verlag, 2016, pp. 52–68. arXiv:1605.04253, doi:10.1007/978-3-319-46475-6_4.

[34] C. Wah, S. Branson, P. Welinder, P. Perona, S. Belongie, The Caltech-UCSD Birds-200-2011 Dataset.

[35] G. Patterson, J. Hays, SUN attribute database: Discovering, annotating, and recognizing scene attributes, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2012, pp. 2751–2758. doi:10.1109/CVPR.2012.6247998.

[36] C. H. Lampert, H. Nickisch, S. Harmeling, Attribute-based classification for zero-shot visual object categorizationa, IEEE Transactions on Pattern Analysis and Machine Intelligence 36 (3) (2014) 453–465. doi:10.1109/TPAMI.2013.140.

[37] Y. Xian, C. H. Lampert, B. Schiele, Z. Akata, Zero-shot learning—a comprehensive evaluation of the good, the bad and the ugly, IEEE Transactions on Pattern Analysis and Machine Intelligence 41 (9) (2019) 2251–2265. doi:10.1109/TPAMI.2018.2857768.

[38] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Vol. 2016-December, IEEE Computer Society, 2016, pp. 770–778. arXiv:1512.03385, doi:10.1109/CVPR.2016.90.

[39] Y. Xian, C. H. Lampert, B. Schiele, Z. Akata, Zero-shot learning – a comprehensive evaluation of the good, the bad and the ugly (2020). arXiv:1707.00600.

[40] M. Norouzi, T. Mikolov, S. Bengio, Y. Singer, J. Shlens, A. Frome, G. S. Corrado, J. Dean, Zero-Shot Learning by Convex Combination of Semantic Embeddings, in: International Conference on Learning Representations, International Conference on Learning Representations, ICLR, 2014. arXiv:1312.5650.

[41] Y. Xian, Z. Akata, G. Sharma, Q. Nguyen, M. Hein, B. Schiele, Latent embeddings for zero-shot classification, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2016. arXiv:1603.08895, doi:10.1109/CVPR.2016.15.

[42] B. Romera-Paredes, P. H. Torr, An embarrassingly simple approach to zero-shot learning, in: 32nd International Conference on Machine Learning, ICML 2015, Vol. 3, International Machine Learning Society (IMLS), 2015, pp. 2142–2151. doi:10.1007/978-3-319-50077-5_2.

[43] S. Changpinyo, W. L. Chao, B. Gong, F. Sha, Synthesized classifiers for zero-shot learning, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2016. arXiv:1603.00550, doi:10.1109/CVPR.2016.575.

[44] Y. Annadani, S. Biswas, Preserving Semantic Relations for Zero-Shot Learning, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2018. arXiv:1803.03049, doi:10.1109/CVPR.2018.00793.

[45] H. Zhang, Y. Long, Y. Guan, L. Shao, Triple Verification Network for Generalized Zero-Shot Learning, IEEE Transactions on Image Processing 28 (1) (2019) 506–517. doi:10.1109/TIP.2018.2869696.

[46] S. Changpinyo, W. L. Chao, B. Gong, F. Sha, Classifier and Exemplar Synthesis for Zero-Shot Learning, International Journal of Computer Vision 128 (1) (2020) 166–201. arXiv:1812.06423, doi:10.1007/s11263-019-01193-1.

[47] Z. Ding, H. Liu, Marginalized latent semantic encoder for zero-shot learning, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Vol. 2019-June, IEEE Computer Society, 2019, pp. 6184–6192. doi:10.1109/CVPR.2019.00635.

[48] H. Zhang, Y. Long, L. Liu, L. Shao, Adversarial unseen visual feature synthesis for Zero-shot Learning, Neurocomputing 329 (2019) 12–20. doi:10.1016/j.neucom.2018.10.043.

[49] H. Huang, C. Wang, P. S. Yu, C.-D. Wang, Generative Dual Adversarial Network for Generalized Zero-shot Learning, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 801–810.

[50] J. Ni, S. Zhang, H. Xie, Dual Adversarial Semantics-Consistent Network for Generalized Zero-Shot Learning, in: Advances in Neural Information Processing Systems 32, NeurIPS 2019, Vol. 32, 2019, pp. 6146–6157.