information

Article

Generalized Zero-Shot Learning for Image Classification—Comparing Performance of Popular Approaches

Elie Saad [1,*], Marcin Paprzycki [2], Maria Ganzha [1], Amelia Bădică [3], Costin Bădică [4], Stefka Fidanova [5], Ivan Lirkov [5] and Mirjana Ivanović [6]

[1] Department of Mathematics and Information Science, Warsaw University of Technology, 46580 Warsaw, Poland
[2] Systems Research Institute, Polish Academy of Sciences, 46580 Warsaw, Poland
[3] Department of Statistics and Business Informatics, University of Craiova, 200585 Craiova, Romania
[4] Department of Computers and Information Technology, University of Craiova, 200585 Craiova, Romania
[5] Institute of Information and Communication Technologies, Bulgarian Academy of Sciences, 1421 Sofia, Bulgaria
[6] Faculty of Sciences, University of Novi Sad, 21000 Novi Sad, Serbia
* Correspondence: elie.saad.stud@pw.edu.pl

Abstract: There are many areas where conventional supervised machine learning does not work well, for instance, in cases with a large, or systematically increasing, number of countably infinite classes. *Zero-shot learning* has been proposed to address this. In generalized settings, the zero-shot learning problem represents real-world applications where test instances are present during inference. Separately, recently, there has been increasing interest in meta-classifiers, which combine the results from individual classifications to improve the overall classification quality. In this context, the purpose of the present paper is two-fold: First, the performance of five state-of-the-art, generalized zero-shot learning methods is compared for five popular benchmark datasets. Second, six standard meta-classification approaches are tested by experiment. In the experiments undertaken, all meta-classifiers were applied to the same datasets; their performance was compared to each other and to the original classifiers.

Keywords: zero-shot learning; generalized zero-shot learning; meta-classifier; performance comparison

check for updates

Citation: Saad, E.; Paprzycki, M.; Ganzha, M.; Bădică, A.; Bădică, C.; Fidanova, S.; Lirkov, I.; Ivanović, M. Generalized Zero-Shot Learning for Image Classification—Comparing Performance of Popular Approaches. *Information* **2022**, *13*, 561. https://doi.org/10.3390/info13120561

Academic Editor: Vincenzo Moscato

Received: 1 September 2022
Accepted: 23 November 2022
Published: 30 November 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

A large number of machine learning methods focus on classifying instances and are applied in cases where the classes have been seen during the model training phase. Here, "seen" means that at least a single (correctly labelled) instance of each existing class was present in the training dataset. The best results of model training are obtained when available labelled datasets are large and when the number of instances in each class is balanced.

However, realistic real-world applications may require the classification of instances belonging to classes that have not been seen before. Here, the examples include, but are not limited to, object recognition (where every object is a class [1]), cross-lingual dictionary induction (where every word is a class [2,3]), and "mind reading", using magnetic resonance imaging scans of the brain (where every word is a class and every instance is a brain scan [4]).

In this context, *zero-shot learning* (ZSL) has been proposed. It is a class of model training approaches that seeks to deal with situations where the classes covered by the training instances and the classes to be classified are disjoint [5–10].

In zero-shot learning, some instances in the feature space are given. In practice, these instances are represented as a vector (usually a real number vector), placed within a feature space (usually a real number space). The set of all instances consists of a subset of instances,

which are labeled, and a second subset of instances, which are not labeled. The classes corresponding to the labeled instances are referred to as *seen* classes, comprising the training set and a part of the testing set. Classes corresponding to unlabeled instances are referred to as *unseen* classes. They constitute the remaining part of the testing set. Note that the set of classes that correspond to the *training instances* and the set of classes that correspond to the *testing instances* may *not* be disjoint. Here, it is reasonably assumed that this more accurately represents real-world circumstances. Zero-shot learning, where it is *not* assumed that the training and testing classes are disjoint, is called *generalized zero-shot learning* (GZSL). It is important to note that GZSL differs from ZSL only in the relationships between the training and the testing datasets.

Machine learning approaches that have been used to solve the ZSL problem have also been applied to GZSL [11]. However, to the best of our knowledge, for both ZSL and GZSL, the proposed model training methods have been evaluated separately. Moreover, even if they have been compared with each other (in a sub-group), different datasets have been used in reported experiments. Finally, even if all datasets and classifiers were the same, only a single performance measure (e.g., Top-1) has been used. However, as shown in [12,13], in the case of ZSL solvers, different performance measures give different answers to the question: Which ZSL approach is the most effective? Therefore, the first goal of this investigation is to experimentally compare the performance of five different approaches to solving the GZSL problem, when the same (the most popular in the literature) benchmark datasets are used. In these comparisons, four different performance measures are applied.

Separately, it can readily be seen (see, for instance, [14]) that the existence of a large number of approaches that can be applied to the same machine learning task results in attempts to develop meta-classifiers. Here, the term "meta-classifier" denotes software that takes as its input results from individual classifiers and combines them, with the intention of obtaining a result that is better than for the best individual classifier. Hence, the second goal of the experimental work undertaken here is to apply standard state-of-the-art meta-classifiers and establish if they can improve performance over individual GZSL approaches.

The remainder of the paper is organized as follows: First, a precise formulation of the problem is provided in Section 2. Next, in Section 3, a brief summary of pertinent literature is provided. A comprehensive description of the experimental setup is provided in Section 4. This is followed by description and analysis of the experimental results obtained in Section 5. Section 6 summarizes the main findings and suggests future research directions.

2. Problem Formulation

The following mathematical formulation of the GZSL problem has been widely accepted in the literature. This definition is geared towards an image processing/classification task, which is the focus of the experimental work reported in this paper. However, after only minimal modifications, it holds in more general settings. Assume the following:

- let $X^s = \{(x_i^s, y_i^s)_{i=1}^{N_s} \in X^s \times T^s\}$ be the set of seen instances
- and let $X^u = \{(x_i^u, y_i^u)_{i=1}^{N_u} \in X^u \times T^u\}$ be the set of unseen instances,
- where $T^s \equiv \{1, \ldots, N^{T^s}\}$ is the set of all seen class labels
- and $T^u \equiv \{N^{T^s} + 1, \ldots, N^{T^s} + N^{T^u}\}$ is the set of all unseen class labels,
- such that $\mathcal{X} = X^s \cup X^u$ is the set of all instances where $X^s \cap X^u = \emptyset$
- and $\mathcal{T} = T^s \cup T^u$ is the set of all class labels where $T^s \cap T^u = \emptyset$ resulting in $N^{T^s} + N^{T^u}$ distinct classes.

Given a dataset of image embeddings and their corresponding class labels $\mathcal{D} = \{(x_i, y_i) \in \mathcal{X} \times \mathcal{T} | i = 1, \ldots, N_{tr} + N_{te}\}$, each image is a real D-dimensional embedding vector comprised of features $x_i \in \mathbb{R}^D$, and each class label is represented by an integer $y_i \in \mathcal{T}$. The following is denoted $\mathcal{X} \stackrel{\text{def}}{=} \mathbb{R}^D$, for generality. The division of the dataset $\mathcal{D}$ results in two subsets: the training and the test sets.

- The training set – is defined as $X^{tr} = \{(x_i^{tr}, y_i^{tr}) \in \mathcal{X} \times T^{tr} | i = 1, \ldots, N_{tr}\}$, such that $y_i^{tr} \in T^{tr} \subset \mathcal{T}$. Let $Y^{tr} = \{y_i^{tr} \in T^{tr} | i = 1, \ldots, N_{tr}\}$ be the set of class labels corresponding to the training set X^{tr}.
- The test set – is defined as $X^{te} = \{(x_i^{te}, y_i^{te}) \in \mathcal{X} \times T^{te} | i = N_{tr} + 1, \ldots, N_{tr} + N_{te}\}$, such that $y_i^{te} \in T^{te} \subset \mathcal{T}$. Let $Y^{te} = \{y_i^{te} \in T^{te} | i = N_{tr} + 1, \ldots, N_{tr} + N_{te}\}$ be the set of class labels to be predicted, corresponding to the set X^{te}.

The goal of generalized zero-shot learning is to deliver a classifier that is trained on the training set X^{tr} and that is able to deliver reasonably good classification performance on the test set X^{te}. It is apparent that the GZSL setting is nearly infeasible without auxiliary information associating the labels of the classes from both the training set and the test set, since some of the training classes are used in conjunction with all of the test classes for testing.

One of the solutions to this problem is to represent each class label y where $1 \leq y \leq N^{T^s} + N^{T^u}$ by its prototype $\pi(y) = p \in \mathcal{P} \stackrel{\text{def}}{=} \mathbb{R}^M$—also known in the literature as its *"semantic embedding"*—where $\pi(\cdot)$ is the prototyping function, and $\mathcal{P}$ is the semantic space, or the semantic embedding space. The resulting prototype vectors, obtained from the prototyping function are obtained in such a way that any two class labels y_0 and y_1 are considered similar, if, and only if, their corresponding prototype representations $\pi(y_0) = p_0$ and $\pi(y_1) = p_1$ are relatively close in the semantic embedding space $\mathcal{P}$. To make this concept more specific, consider the following example: the inner product of the resulting similar prototype vectors in $\mathcal{P}$ is large, i.e., $\langle \pi(y_0), \pi(y_1) \rangle_{\mathcal{P}}$ is large. Prototyping all of the class labels into a joint semantic space, i.e., $\{\pi(y) | y \in T^{tr} \cup T^{te}\}$, forces the labels to become related. In consequence, this alleviates the prior problem of having nearly disjoint class sets, so that the model can learn from the training set and test its predictions from the test set.

3. Related Work

Multiple approaches have been proposed to solve the ZLS/GZSL problem. An extended state-of-the-art summary can be found in [13]. Here, we focus our attention mostly on work directly related to the GZSL problem. Moreover, since only standard state-of-the art meta-classifiers are experimented with, their detailed description is omitted (this can be found in [13]). Details concerning their implementation are summarized in Section 4.2.

There are currently five classes of approaches to the solution of the ZSL/GZSL problem [11].

1. The first utilizes a linearity factor, i.e., it is based on a learning function that is linear. Here, for instance, deep visual semantic embedding (*DeViSE*) [15], attribute label embedding (*ALE*) [16], and structured joint embedding (*SJE*) [17] algorithms use a bilinear compatibility function, i.e., a symmetrically bilinear mapping, in which the *stochastic gradient descent* [18] method is used for learning and is implicitly regularized by early stopping. The "embarrassingly simple approach to zero-shot learning" approach (*ESZSL*) [19] uses the square loss error, to learn the bilinear compatibility function between image embedding and the class values, i.e, class labels, and explicitly defines regularization, which regularizes the unregularized risk minimization formulation with respect to the Frobenius norm. In the semantic autoencoder (*SAE*) model, the training instances are projected into the semantic embedding space $\mathcal{P}$ with the projection matrix W, and then projected back into the feature space $\mathcal{X}$, with a projection matrix W^*, where W^* is the conjugate transpose.
2. The second group of approaches adds a non-linearity component to the linear compatibility function. Here, the latent embeddings method (*LATEM*) [20] learns a piecewise linear compatibility and uses the same ranking loss function as the *DeViSE* approach, combined with stochastic gradient descent. *LATEM* learns multiple mappings corresponding to different visual characteristics of the data and uses a latent variable to select which matrix mapping to use in predictions. The cross modal transfer (*CMT*) method, described in [21], trains a two-layer neural network using a non-linear mapping function tanh.

3. The third group of approaches uses probabilistic embeddings mapping to the class values [5]. Here, the direct attribute prediction (*DAP*) algorithm makes class predictions by combining scores of the learned attribute classifiers, whereas the indirect attribute prediction (*IAP*) algorithm approximates the probabilities of the attributes associated with the training instances by predicting the probabilities of each training class value.
4. The fourth group of approaches expresses the input image features and the semantic embeddings as a mixture of seen classes. Here, the semantic similarity embedding (*SSE*) method [22] connects class values within both the semantic-embedding space and the feature-embedding space of the feature vectors. The convex combination of the semantic embeddings (*ConSE*) [23] method learns the probability of a feature vector belonging to a specific class. The synthesized classifiers (*SynC*) [24] approach first learns a mapping from the semantic-embedding space to the model space, which holds the classifiers associating the real class values and what are referred to as phantom class values. The class values, along with the set of phantom class values, which are introduced to connect the seen and unseen class values, form a weighted bipartite graph. The objective of *SynC* is to align the semantic-embedding space and the model space, resulting in reducing the distortion error. The generative framework for the zero-shot learning (*GFZSL*) [25] method applies generative modeling. Here, each class-conditional distribution is modeled as a multivariate Gaussian distribution.
5. Finally, in the fifth set of models, during training, both seen and unseen classes are included in the training data. The discriminative semantic representation learning (*DSRL*) method [26] can be applied to any method that is based on a compatibility learning function, i.e., a mapping of the image embeddings to class labels. It learns the feature-embedding vectors from the training instances with non-negative matrix factorization and aligns the feature-embedding vectors with the semantic-embedding vectors of their corresponding class values.

Taking into account the results of the literature review, and the goals of the project that this work is a part of, the following question is raised: Which classifiers should be experimented with? While testing them all is tempting, it would require a lot of resources and would not fit into a single report. Upon further reflection, it was decided that testing representatives of all five groups of approaches (e.g., one per group) could result in a lot of questions concerning the selection process and the fairness of comparison (since each class of approaches addresses the ZSL/GZSL problem from a very different perspective). Therefore, it was decided that a reasonable first step would be to comprehensively explore approaches belonging to the first class, where a large number of methods have already been proposed. Hence, five individual classifiers belonging to the first class of approaches and implementing different linear factors, i.e., DeViSE, ALE, SJE, ESZSL and SAE, were chosen. This decision was supported by the fact that all these approaches have been explored in [11], which can be treated as the basis on which a further, relatively fair, comparison of results can be made.

4. Experimental Setup

In the following sections, the key aspects of the experimental setup are discussed, including: (1) implementation details of the ZSL/GZSL algorithms, (2) implementation details of the meta-classifiers, (3) performance measures applied to the obtained results, and (4) the datasets used in the experiments.

4.1. Individual Classifier Implementations

Individual classifiers were taken from the sources reported in the respective publications (see Section 3). Details concerning their implementation have been reported in [12,13]. In the context of GZSL, before proceeding with the actual experiments, light hyperparameter tuning was performed. Specifically, for each classifier and for each dataset (see Section 4.4), a series of ten experiments was performed. Each time the hyperparameters

were slightly shifted in a direction where better results would be anticipated (see also [12,13]). In these experiments, the Top-1 accuracy measure was used (see Section 4.3) to assess performance. The final values of the hyperparameters for each of the individual classifiers and for each dataset are summarized in Table 1. Here, it must be stressed that, while these hyperparameter values delivered the best accuracy for the experiments performed, it is not claimed that they are the optimal values. It is possible, and quite likely, that there exist hyperparameters that would result in better performance for each individual classifier, for each dataset and, as will be shown, for each separate performance measure. However, searching the complete hyperparameter space was beyond the scope of this investigation. Instead, it was assumed that a reasonably good set of hyperparameters should be applied for each classifier and dataset and used to compare classifier performance for the performance measures considered.

Table 1. Hyperparameters used in experiments for each individual classifier and dataset.

CLF	CUB	AWA1	AWA2	aPY	SUN
DeViSE	lr $= 1$ mr $= 1$ norm $= L^2$	lr $= 0.01$ mr $= 200$ norm $=$ std	lr $= 0.001$ mr $= 150$ norm $=$ std	lr $= 1$ mr $= 1$ norm $= L^2$	lr $= 0.01$ mr $= 3$ norm $=$ None
ALE	lr $= 0.3$ norm $= L^2$	lr $= 0.01$ norm $= L^2$	lr $= 0.01$ norm $= L^2$	lr $= 0.04$ norm $= L^2$	lr $= 0.1$ norm $= L^2$
SJE	lr $= 0.1$ mr $= 4.0$ norm $=$ std	lr $= 1$ mr $= 2.5$ norm $= L^2$	lr $= 1$ mr $= 2.5$ norm $= L^2$	lr $= 0.01$ mr $= 1.5$ norm $=$ None	lr $= 1$ mr $= 2$ norm $=$ std
ESZSL	$\gamma = 3$ $\lambda = 0$	$\gamma = 3$ $\lambda = 0$	$\gamma = 3$ $\lambda = 0$	$\gamma = 2$ $\lambda = 0$	$\gamma = 3$ $\lambda = 2$
SAE	$\lambda_1 = 80$ $\lambda_2 = 0.2$	$\lambda_1 = 3.2$ $\lambda_2 = 0.8$	$\lambda_1 = 0.8$ $\lambda_2 = 0.2$	$\lambda_1 = 0.16$ $\lambda_2 = 2.56$	$\lambda_1 = 0.32$ $\lambda_2 = 0.08$

In Table 1, *lr* denotes the learning rate; λ_1 and λ_2 are the weighting coefficients of the Sylvester equation for the encoder and the decoder, respectively. The image embedding normalization functions that have been used are: the Euclidean norm denoted as L^2 and the standardization of features, which changes the mean and scales to unit variance. The latter, for the training instance vector x, is defined as follows:

$$||x||_{\text{std}} = \frac{x - \bar{x}}{s(x)}, \tag{1}$$

where $\bar{x}$ is the mean of the training instances, defined as

$$\bar{x} = \frac{\sum_{i=1}^{N_{tr}} x_i}{N_{tr}}, \tag{2}$$

and $s(\cdot)$ is the standard deviation of the training instances, defined as follows:

$$s(x) = \sqrt{\frac{\sum_{i=1}^{N_{tr}} (x_i - \bar{x})^2}{N_{tr}}}. \tag{3}$$

Finally, *mr* is the margin value, which is used by each individual classifier. It applies to the stochastic gradient descent approach for optimization in *DeViSE*, *ALE* and *SJE*.

4.2. Meta-Classifier Implementations

Six standard state-of-the-art meta-classifiers were tested, namely, meta-decision tree (*MDT*) [27], deep neural network (*DNN*) [28], a game-theory-based approach (*GT*) [29], an auction-based model (*Auc*) [29], a consensus-based approach (*Con*) [30], and a simple ma-

jority voting approach (*MV*) [31]. Here, *GT*, *Auc*, *Con* and *MV* classifiers were implemented exactly as described in the cited literature.

The implementation of the *MDT* differs from that described in [27] by not applying the *weight* condition on the individual classifiers. The weight condition is defined as a function, which returns the fraction of the training examples that have been used by the individual classifier to estimate the class distribution for a given training instance. Since, in the case of the experiments performed, each individual classifier uses the entirety of the dataset, weighting is irrelevant.

Finally, a simple neural network was implemented for the *DNN*. It included two hidden layers using the rectified linear activation function. The stochastic gradient descent algorithm was applied as an optimization function for the network, where the mean squared error loss function was used.

Again, as in the case of individual classifiers, the goal was not to create the most powerful meta-classifier for a given context, but to implement standard state-of-the-art meta-classifiers and reflect on their potential in the context of GZSL. Therefore, the same approach that was used to obtain the hyperparameters for the individual classifiers (see [12,13] and Section 4.1), was applied to obtain the *MDT* hyperparameters, which are summarized in Table 2.

Table 2. Meta Decision Tree Hyperparameters.

Hyperparameter	CUB	AWA1	AWA2	aPY	SUN
α	1	0.013	0.185	0.0101	3.21
β	1	0.16632	0.185	0.159171	3.21

The α hyperparameter is the maximum displacement value and the β hyperparameter is the entropy displacement value. The learning rate of the *DNN* meta-classifier was set to 1×10^{-4}.

The implementation of both the individual and the meta-classifiers was carried out from scratch; the resulting code that was used in the experiments can be found in the Github repository (https://github.com/Inars/Zero-Shot_Learning_for_Object_Recognition_and_Classification, accessed on 1 July 2020).

4.3. Performance Measures

In all the experiments undertaken, the accuracy of the classifiers was assessed using four accuracy measures: Top-1 (T1), Top-5 (T5), logarithmic loss (LogLoss) and the F1 score (F1).

The T1 accuracy, as described in [11], was measured for the single label image classification. In this case, the prediction is accurate whenever the predicted class corresponds to the ground truth. However, for the ZSL/GZSL the concern is with obtaining high performance, even for sparsely populated classes. Therefore, the T1 accuracy was measured by averaging the class accuracy of each class independently, in the following way:

$$\text{acc}_\mathcal{Y} = \frac{1}{N^{T^s} + N^{T^u}} \sum_{c=1}^{N^{T^s}+N^{T^u}} \frac{\#\text{ correct predictions in } c}{\#\text{ samples in } c} \tag{4}$$

The T5 accuracy was measured in the same way as the T1 accuracy. Here, the prediction was considered correct whenever one of the top five predicted classes was true. In the context of ZSL/GZSL, the T1 accuracy can be seen as a measure of exact precision, while the T5 accuracy measures the robustness of the approach.

Next, the F1 score [32], which calculates the precision and robustness, i.e., the recall of the model, was used. Finally, since this research focuses on generalized zero-shot learning, where the search space at evaluation time is composed of training and test classes (where $c_i \in \mathcal{T}$), a further accuracy measure is typically applied; specifically, the harmonic mean of

the training accuracy and the test accuracy is computed. The harmonic mean is selected, instead of the arithmetic mean, because the harmonic mean is not significantly affected by the potentially large difference(s) between the training accuracy and the testing accuracy [33]. The harmonic mean can be seen as a way to capture variance in the differences.

Note that most predictions of the models are not probabilistic; therefore, the prediction values were converted into a probabilistic distribution by applying the normalized exponential function [34], also known as the *softmax* function. The normalized exponential function σ is a function that normalizes the input vector $z = (z_1, \ldots, z_k)^T \in \mathbb{R}^k$ into a probability distribution consisting of k probabilities.

4.4. Datasets Used in Experiments

From the many datasets found in the literature in the context of the GZSL problem, five of the most widely used were selected for this study, namely: Caltech-UCSD-Birds 200-2011 (CUB) [35], Scene UNderstanding (SUN) [36], Animals with Attributes 1 (AWA1) [5], Animals with Attributes 2 (AWA2) [11], and Attribute Apascal&Ayahoo (aPY) [37].

Each of these five datasets forms a continuous attribute space. All the datasets consist of image embeddings. In the experiments undertaken, the dataset splitting followed the suggestions in [11], as represented in Table 3. Here the scalars N_{tr}^{te} and N_{te}^{te} denote the number of instances used during the testing phase; the type of dataset is displayed in the "detail" column, and the number of semantic attributes is displayed in the "M" column.

Table 3. Statistics for the Datasets.

DS	Detail	M	$N^{Ts} + N^{Tu}$	$\lvert T^{tr} \rvert$	$\lvert T^{te} \rvert$	$N_{tr} + N_{te}$	N_{tr}	N_{tr}^{te}	N_{te}^{te}
CUB	fine	312	200	$100 + 50$	50	11,788	7057	1764	2967
AWA1	coarse	85	50	$27 + 13$	10	30,475	19,832	4958	5685
AWA2	coarse	85	50	$27 + 13$	10	37,322	23,527	5882	7913
aPY	coarse	64	32	$15 + 5$	12	15,339	5932	1483	7924
SUN	fine	102	717	$580 + 65$	72	14,340	10,320	2580	1440

The **aPY** dataset consists of two datasets, the **Pascal dataset** and the **Yahoo dataset**, merged into one. It is a coarse-grained dataset consisting of 32 classes. Of the 32 classes available, 20 classes provided from the **Pascal dataset** are used for training. The 12 classes, provided by the **Yahoo dataset** are used for testing. Of the 20 classes from the **Pascal dataset**, five classes are selected randomly to serve as validation classes. The **aPY** dataset consists of 64 attributes, describing the classes forming the semantic embedding space.

The **AWA1** dataset is a coarse-grained dataset, consisting of 50 classes. Of these 50 classes, 40 classes are used for training, with 13 randomly selected for validation. The remaining 10 classes are used for testing. The **AWA1** dataset consists of 85 attributes. In terms of the number of image instances, the **AWA1** dataset consists of a total of 30,475 image instances.

Like **AWA1**, **AWA2** was introduced in [11] for use as an alternative, since the images on the **AWA1** dataset are not available for public use. The **AWA2** dataset is publicly available. Here, recall that in this investigation the images themselves were not actually needed, as only the features that were extracted from them were used. This is why we were able to experiment with both **AWA1** and **AWA2** datasets. **AWA2** is a coarse-grained dataset, consisting of 50 of the same classes used in the **AWA1** dataset. The **AWA2** dataset consists of the same 85 attributes used in the **AWA1** dataset. Unlike **AWA1**, the **AWA2** dataset consists of 37,322 image instances from public web sources, protected under a free-use and redistribution licence, none of which overlap with the images from the **AWA1** dataset. In a similar way to how the **AWA1** dataset is split, of the 50 available classes in **AWA2**, 40 classes are used for training, with 13 randomly selected for validation and the remaining 10 classes used for testing.

The **CUB** dataset is a fine-grained dataset in terms of both the number of classes and the number of images and consists of 200 classes of bird species and 11,788 bird images. Of

the 200 available classes, 150 classes are used for training, with 50 randomly selected for validation and the remaining 50 classes used for testing. The classes in the **CUB** dataset are annotated with 312 attributes forming the continuous attribute space.

The **SUN** dataset is a fine-grained dataset in terms of both the number of classes and the number of images available. It consists of 717 classes of different scenes and comprises a total of 14,340 images. Of the 717 available classes, 645 classes are used for training, with 65 randomly selected for validation and the remaining 72 classes used for testing. The classes in the **SUN** dataset are annotated with 102 attributes forming the continuous attribute space.

All five datasets used in the experimental procedure already include the feature vectors, which contain the features extracted from the image instances provided within each dataset. Since feature extraction to form the feature space is not the focus of this investigation, it was assumed that the datasets provided in [11] were sufficient. Therefore, various feature extraction techniques, such as ResNet, were not applied (e.g., as a preprocessing step) to the images provided in the datasets.

5. Experimental Results and Their Analysis

As stated above, the first goal of the experiments undertaken was to evaluate and compare the behavior of the five individual classifiers. The second goal was to explore the performance of the six meta-classifiers, including a comparison of their performance with the individual classifiers.

5.1. Experiments with Individual Classifiers

The first set of results represents the T1 accuracy and is summarized in Table 4. Here, as defined in Section 2, Y^{tr} represents the set of training class labels, Y^{te} denotes the set of testing class labels, while $\hat{H}$ represents the harmonic mean of the two (as described in Section 4.3).

The first group of rows copies results found in [11] (where only the T1 measurement was used to assess the performance), whereas the second group of rows displays the results obtained during the experiments undertaken (in-house implementation). Moreover, the first row of the results reported for the *SAE* classifier for the in-house implementation represents the accuracy obtained from the encoder, whereas the second row represents the accuracy obtained from the decoder. Finally, for each classifier and for each dataset, the "best" results are marked in bold font.

Table 4. Individual Classifier Results for the Top-1 Accuracy.

DS	**CUB**			**AWA1**			**AWA2**			**aPY**			**SUN**		
CLF	$f(Y^{tr})$	$f(Y^{te})$	$\hat{H}$	$f(Y^{tr})$	$f(Y^{te})$	$\hat{H}$	$f(Y^{tr})$	$f(Y^{te})$	$\hat{H}$	$f(Y^{tr})$	$f(Y^{te})$	$\hat{H}$	$f(Y^{tr})$	$f(Y^{te})$	$\hat{H}$
Results reported in [11]															
DeViSE	23.8	52	32.8	13.4	68.7	22.4	17.1	74.7	**27.8**	3.5	78.4	6.7	16.9	27.4	20.9
ALE	23.7	62.8	**34.4**	16.8	76.1	**27.5**	14	81.8	23.9	4.6	73.7	**8.7**	21.8	33.1	**26.3**
SJE	23.5	59.2	33.6	11.3	74.6	19.6	8	73.9	14.4	1.3	71.4	2.6	14.4	29.7	19.4
ESZSL	14.7	56.5	23.3	6.6	75.6	12.1	5.9	77.8	11	2.4	70.1	4.6	11	27.9	15.8
SAE	7.8	54	13.6	1.8	77.1	3.5	1.1	82.2	2.2	0.4	80.9	0.9	8.8	18	11.8
In-House Implementation															
DeViSE	23.41	61.96	33.98	18.01	84.41	**29.69**	17.3	71.78	**27.88**	4.12	76.28	7.8	18.54	32.75	23.68
ALE	26.07	62.74	**36.83**	13.46	80.14	23.05	12.15	77.59	21.01	10.26	69.85	**17.9**	23.68	37.13	**28.92**
SJE	22.82	60.8	33.19	9.93	79.29	17.64	10.48	78.82	18.5	6.26	73.31	11.54	18.75	33.37	24.01
ESZSL	14.7	56.53	23.34	5.29	86.84	9.98	4.04	88.83	7.72	2.25	81.07	4.39	13.75	28.41	18.53
SAE	13.86	49.88	21.69	5.29	80.52	9.92	5	81.42	9.42	8.28	27.97	12.77	16.81	24.69	20
	15.72	57.02	24.64	14.72	82.93	25	13.86	87.2	22.41	9.48	56.62	16.24	19.03	31.2	23.64

The following observations can be made based on the results presented in Table 4, where experiments based on in-house implementation are compared with the results presented in [11]. Note that, if the difference between the results is within ±5% they are

considered to be close enough to be considered indistinguishable. This is stipulated since, as stated earlier, the aim of this investigation was not to fine-tune the hyperparameters, but to fairly compare the performance of different approaches to GZSL.

- In-house implementation of *DeViSE*—obtained a 7.29% increase in accuracy (over results reported in [11]) for the **AWA1** dataset.
- *ALE*—achieved a 9.2% increase in accuracy for the **aPY** dataset.
- *SJE*—delivered an 8.98% increase in accuracy for the **aPY** dataset.
- *ESZSL*—produced similar results on all five datasets.
- *SAE*—achieved an 8.09%, a 6.42%, a 7.22%, an 11.87% and an 8.2% increase in accuracy for the **CUB**, **AWA1**, **AWA2**, **aPY** and **SUN** datasets, respectively.

In summary, the results for in-house implementation were of slightly better performance than the results reported in [11]. However, the results were relatively close (within 10%) and the difference probably originated from differences in hyperparameter tuning. Overall, these results support each other as representing a fair estimate of the current state-of-the-art accuracy of GZSL solvers for the T1 performance measure.

With respect to the results originating from the in-house implementation of individual classifiers (see Section 4.1) reported in the bottom part of Table 4, it can be seen that: (i) *ALE* produced the highest accuracy value on the **CUB**, **SUN** and **aPY** datasets (38.83%, 17.9% and 28.92%, respectively), whereas *DeViSE* achieved the highest accuracy values on both the **AWA1** and **AWA2** datasets (29.69% and 27.88%, respectively); (ii) none of *SJE*, *ESZSL* or *SAE* outperformed the remaining classifiers for any of the five datasets; (iii) for the harmonic means results, only *DeViSE*, *ALE* and *SJE* were above 30% for the **CUB** dataset; (iv) all of the individual classifiers performed considerably worse on the **aPY** dataset than on the other datasets. This result is consisten with that reported in [12,13] for the ZSL settings. It also corresponds to the pattern of results reported in [11]; and finally, (v) there was a significant drop in accuracy across the board compared to the results from the ZSL experiments reported in [12,13]. The reason for this drop was the availability of the seen classes taken from the source domain T^{tr} during the testing phase in the target domain T^{te}. In other words, the fact that $T^{tr} \cap T^{te} \neq \emptyset$. Here, the seen classes acted as "deterrence" for the individual classifiers, since they were trained to exclusively classify the seen classes from the source domain, while no unseen class from the target domain was present during the training phase. This gave the seen classes "an edge" over their counterparts, i.e., an increased bias during the prediction process. This resulted in the seen classes having a higher selection rate than their unseen counterparts. As a consequence, this reduced the overall accuracy of the individual classifiers.

The remaining experimental results, reported below, could not have been compared to others, as, to the best of our knowledge, no fully comparable results obtained for the same approaches, with the same datasets, with the same performance measures, exist in the literature. The first set of results was obtained for the T5 accuracy and is reported in Table 5. Here, again, the "best" results are marked in bold font.

Table 5. Individual Classifier Results for the T5 Accuracy.

DS	**CUB**			**AWA1**			**AWA2**			**aPY**			**SUN**		
CLF	$f(Y^{tr})$	$f(Y^{te})$	$\hat{H}$	$f(Y^{tr})$	$f(Y^{te})$	$\hat{H}$	$f(Y^{tr})$	$f(Y^{te})$	$\hat{H}$	$f(Y^{tr})$	$f(Y^{te})$	$\hat{H}$	$f(Y^{tr})$	$f(Y^{te})$	$\hat{H}$
DeViSE	59.6	87.6	70.9	69.3	98.0	**81.2**	66.9	96.8	**79.1**	58.3	93.0	**71.7**	47.3	62.8	54.0
ALE	65.6	89.5	**75.7**	23.1	80.1	60.0	62.7	97.8	76.4	54.2	98.1	69.8	57.1	69.2	**62.6**
SJE	62.8	86.7	72.9	66.0	96.6	78.4	66.3	97.0	78.8	51.0	93.3	65.9	48.3	64.1	55.1
ESZSL	58.7	86.5	69.9	63.9	97.8	77.3	63.0	98.4	76.8	40.9	97.8	57.7	42.5	58.4	49.2
SAE	13.9	51.9	21.8	15.1	83.2	25.5	14.6	84.0	24.8	16.6	36.5	22.8	19.0	25.0	21.6
	15.7	57.8	24.7	23.1	84.6	36.3	21.3	88.4	34.3	17.8	57.9	27.2	21.0	31.5	25.2

The following observations follow from the results reported in Table 5: (A) *DeViSE* produced the highest performance on the **AWA1**, **AWA2** and **aPY** datasets (81.22%, 79.09% and 71.65%, respectively); (B) *ALE* achieved the best overall performance on both the **CUB**

and **SUN** datasets (75.71% and 62.57%, respectively); (C) all of the reported results were less than 80%; (D) The **SUN** dataset appeared to be the "hardest" dataset when the T5 performance measure was applied; finally, (E) *SAE* performed the worst. Most of the values of *SAE* for the T5 measure were relatively similar to one another across all five datasets. This should be taken into account when comparing the values of *SAE* for the T5 measurement to the values for the T1 measure.

The next set of results relate to the performance of the five classifiers, measured according to LogLoss accuracy. The results are summarized in Table 6. Note that, here, the lowest value is the "best"; the best results are marked in bold font).

Table 6. Individual Classifier Results for the LogLoss Accuracy.

DS	**CUB**			**AWA1**			**AWA2**			**aPY**			**SUN**		
CLF	$f(Y^{tr})$	$f(Y^{te})$	$\hat{H}$	$f(Y^{tr})$	$f(Y^{te})$	$\hat{H}$	$f(Y^{tr})$	$f(Y^{te})$	$\hat{H}$	$f(Y^{tr})$	$f(Y^{te})$	$\hat{H}$	$f(Y^{tr})$	$f(Y^{te})$	$\hat{H}$
DeViSE	5.24	5.23	5.23	3.48	3.00	3.22	3.38	2.95	3.15	3.47	3.31	3.39	6.53	6.53	6.53
ALE	5.29	5.29	5.29	3.90	3.90	3.90	3.91	3.91	3.91	3.56	3.44	3.50	6.57	6.57	6.57
SJE	4.90	4.40	4.62	3.78	3.78	3.78	3.81	3.77	3.79	3.47	3.47	3.47	6.88	6.48	6.67
ESZSL	3.78	3.43	**3.59**	2.93	2.18	**2.50**	2.92	2.13	**2.46**	4.67	1.79	**2.57**	5.63	5.57	**5.60**
SAE	4.98	4.92	4.95	3.69	3.51	3.6	3.67	3.46	3.56	3.38	3.03	3.20	6.21	6.18	6.19
	5.05	5.01	5.03	3.66	3.51	3.58	3.63	3.43	3.52	3.5	2.87	3.15	6.23	6.19	6.21

Comparing the results in Table 6, the following observations can be made: (a) *ESZSL* achieved the lowest values on both the **AWA1** and **aPY** datasets (1.41 and 2.3, respectively); (b) *ESZSL* achieved the lowest values for each of the **CUB**, **AWA1**, **AWA2**, **aPY** and **SUN** datasets (3.59, 2.5, 2.46, 2.57 and 5.6, respectively). When evaluated from the LogLoss perspective, the **SUN** dataset again appeared to be the most difficult to deal with.

Finally, in Table 7, the performance of the five individual classifiers is compared in terms of the F1 measure. Here, the "best" (highest) values are marked in bold font.

Table 7. Individual Classifier Results for the F1 Accuracy.

DS	**CUB**			**AWA1**			**AWA2**			**aPY**			**SUN**		
CLF	$f(Y^{tr})$	$f(Y^{te})$	$\hat{H}$	$f(Y^{tr})$	$f(Y^{te})$	$\hat{H}$	$f(Y^{tr})$	$f(Y^{te})$	$\hat{H}$	$f(Y^{tr})$	$f(Y^{te})$	$\hat{H}$	$f(Y^{tr})$	$f(Y^{te})$	$\hat{H}$
DeViSE	0.24	0.62	0.34	0.15	0.88	**0.26**	0.09	0.80	**0.16**	0.03	0.76	0.06	0.19	0.33	0.24
ALE	0.26	0.62	**0.37**	0.12	0.87	0.22	0.07	0.86	0.13	0.05	0.74	**0.10**	0.24	0.37	**0.29**
SJE	0.23	0.6	0.33	0.1	0.86	0.18	0.07	0.86	0.13	0.03	0.74	0.06	0.19	0.33	0.24
ESZSL	0.15	0.56	0.23	0.07	0.9	0.13	0.04	0.92	0.08	0.01	0.83	0.02	0.14	0.28	0.19
SAE	0.26	0.50	0.22	0.06	0.83	0.11	0.05	0.85	0.09	0.05	0.23	0.09	0.17	0.25	0.20
	0.26	0.59	0.36	0.28	0.86	0.43	0.21	0.90	0.35	0.05	0.70	0.10	0.31	0.32	0.31

The following observations can be made on the basis of the results reported in Table 7: (1) The *ALE* algorithm achieved the highest performance on the **CUB**, **aPY** and **SUN** datasets (0.37, 0.1, and 0.29, respectively); (2) *DeViSE* achieved the highest values on the **AWA1** and **AWA2** datasets (0.26 and 0.16, respectively); and (3) the **aPY** dataset was the hardest to deal with when the F1 accuracy measure was applied.

Overall, on the basis of all the experiments performed, similar conclusions can be drawn to those reported in [12,13]: (i) Different performance measures promoted different GZSL approaches; (ii) the **aPY** and **SUN** datasets were the most difficult to classify, depending on the performance measure that was being used. This differed from the ZSL settings, where only the **aPY** dataset was found to be difficult; and (iii) none of the individual classifiers can be considered "the best". Moreover, none of the classifiers delivered particularly good results, regardless of the dataset and performance measure used to evaluate it.

The concern to identify the best overall approach was addressed in [13] by introducing a competitive scoring scheme. Specifically, each of the five classifiers was assigned scores from 5 to 1 for each dataset for each performance measure, depending on its result (the best performance received 5 points, while the worst received 1 point). Next, the results were

added. The same approach to representing "robustness" for all classifiers was applied; the results are displayed in Table 8. The "best" results for each dataset, and overall, are marked in bold.

Table 8. Individual Classifier Combined Performance.

CLF	CUB	AWA1	AWA2	aPY	SUN	Total
DeViSE	13	**19**	**19**	14	13	**78**
ALE	**16**	11	11	**15**	**17**	70
SJE	14	12	13	12	13	64
ESZSL	11	12	11	10	10	54
SAE	6	6	9	12	10	43

The observations from Table 8 can be summarized as follows: (A) *DeViSE* performed best for the **AWA1** and **AWA2** datasets and also obtained the best overall result (78 points); (B) *ALE* performed the best for the **CUB**, **aPY**, and **SUN** datasets; and (C) *SAE* performed the worst for almost all datasets, as well as overall.

These results differ from these reported in [12,13], where the best overall score was reported for the *ESZSL* classifier. Overall, for the GZSL problem, if specific characteristics of the dataset are not known beforehand, the *DeViSE* approach may be the one to try first. However, the results obtained strengthen the view that much more work is needed to develop a deeper understanding of the relationships between datasets, approaches and performance measures with respect to the GZSL problem.

5.2. Performance of Meta-Classifiers

The second part of the investigation concerns the experimental evaluation of the performance of meta-classifiers. Here, the T1 accuracy results are summarized in Table 9. The "best" results are marked in bold font.

Table 9. Meta-Classifier Results for the T1 Accuracy.

DS	CUB			AWA1			AWA2			aPY			SUN		
CLF	$f(Y^{tr})$	$f(Y^{te})$	$\hat{H}$	$f(Y^{tr})$	$f(Y^{te})$	$\hat{H}$	$f(Y^{tr})$	$f(Y^{te})$	$\hat{H}$	$f(Y^{tr})$	$f(Y^{te})$	$\hat{H}$	$f(Y^{tr})$	$f(Y^{te})$	$\hat{H}$
MV	21.98	66.83	33.08	8.76	84.01	15.87	7.34	81.61	13.48	3.24	80.36	6.23	22.64	37.6	**28.26**
MDT	25.96	75.89	38.69	20.21	80.01	32.27	20.4	80	32.51	38.53	62.65	47.72	11.02	90.01	19.63
DNN	25.9	75.89	38.62	20.18	80.05	32.23	20.46	80.02	32.59	38.53	64.17	48.15	11.16	90.06	19.86
GT	26.45	75.49	**39.17**	20.86	80.59	**33.14**	21.03	80.24	33.33	38.62	64	**48.17**	10.36	90.06	18.58
Con	25.97	75.71	38.67	20.18	80.05	32.23	21.42	80.3	**33.82**	37.92	64.84	47.85	10.91	90.08	19.47
Auc	25.96	75.89	38.69	20.18	80.05	32.23	20.46	80.03	32.59	38.53	62.65	47.72	10.77	90.03	19.25

The following key information can be derived from the results reported in Table 9: (i) The *GT* achieved the highest performance for the **CUB**, **AWA1** and **aPY** datasets (39.17%, 33.14% and 48.17%, respectively); (ii) *Con* achieved the highest overall T1 accuracy for the **AWA2** dataset (33.82%); (iii) *MV* achieved the highest T1 accuracy value for the **SUN** dataset (28.26%); (iv) these results imply that none of *MDT*, *DNN* or *Auc* achieved the best scores for any single dataset; (v) all the results were below 50% accuracy; (vi) all the results obtained were higher than those reported in Table 4; (vii) the largest difference was for the *ESZSL* classifier and the **AWA2** dataset (69.07%), while the smallest was for the *SAE* classifier and the **CUB** dataset (0.12%); and (viii) the **SUN** dataset was found to be the most difficult when the performance of the meta-classifiers was measured in terms of the T1 accuracy measure.

It is important to note that the T5 accuracy was not reported, as some meta-classifiers, e.g., *DNN*, returned the individual classifier result as an output. Hence, since the experimentation was undertaken for five individual classifiers, the resulting accuracy of the meta-classifier using the T5 measure would always be 100%, regardless of the correctness of the output.

For use of the F1 accuracy measure to assess the performance of the meta-classifiers, the results are summarized in Table 10 (bold-font-marked results are "the best").

Table 10. Meta-Classifier Results for the F1 Accuracy.

DS	**CUB**			**AWA1**			**AWA2**			**aPY**			**SUN**		
CLF	$f(Y^{tr})$	$f(Y^{te})$	$\hat{H}$	$f(Y^{tr})$	$f(Y^{te})$	$\hat{H}$	$f(Y^{tr})$	$f(Y^{te})$	$\hat{H}$	$f(Y^{tr})$	$f(Y^{te})$	$\hat{H}$	$f(Y^{tr})$	$f(Y^{te})$	$\hat{H}$
MV	0.66	0.22	**0.33**	0.89	0.09	0.17	0.88	0.06	0.1	0.81	0.02	0.03	0.38	0.23	**0.28**
MDT	0.12	0.36	0.19	0.01	0.02	0.02	0.02	0.02	0.02	0.1	0.02	0.03	0.11	0.1	0.1
DNN	0.12	0.36	0.18	0.02	0.01	0.02	0.02	0.02	0.02	0.11	0.67	**0.19**	0.12	0.1	0.11
GT	0.16	0.2	0.18	0.1	0.3	0.2	0.12	0.01	0.1	0.11	0.12	0.11	0.03	0.1	0.05
Con	0.12	0.29	0.17	0.28	0.39	**0**.32	0.16	0.11	**0.13**	0.05	0.17	0.07	0.1	0.12	0.11
Auc	0.19	0.36	0.19	0.01	0.02	0.02	0.02	0.03	0.02	0.11	0.02	0.03	0.12	0.1	0.11

The following observations can be derived from Table 10: (a) *MV* achieved the highest values for the **CUB** and **SUN** datasets (0.33 and 0.28, respectively); (b) *Con* achieved the highest performance for the **AWA1** and **AWA2** datasets (0.32 and 0.13, respectively); (c) *DNN* achieved the highest accuracy score for the **aPY** dataset (0.19); (d) none of *MDT*, *GT* or *Auc* achieved the best accuracy score for any dataset; (e) all of the reported results were below 0.4; (f) the obtained results were comparable to, but somewhat worse than, the results reported in Table 7; and (g) the **AWA2** dataset was the most difficult using the F1 accuracy measure.

Comparing the results obtained when applying the F1 accuracy score to the meta-classifiers with those obtained for the individual classifiers, it can be seen that the meta-classifiers performed better than the individual classifiers on the **AWA1** and **aPY** datasets (0.32 and 0.28 compared to 0.26 and 0.1, respectively). At the same time, the individual classifiers obtained better results on the **CUB**, **AWA2**, and **SUN** datasets (0.37, 0.17 and 0.29 compared to 0.33, 0.13 and 0.28, respectively).

Using the method of competitive point distribution described above to measure the combined performance of the individual classifiers, the results calculated for the meta-classifiers are reported in Table 11 ("best" results reported in bold font). Both the performance measures, T1 and F1, were combined for the meta-results. The top scorer, in a given category, is given six points, since there are six meta-classifiers.

Table 11. Meta-Classifier Combined Performance.

CLF	**CUB**	**AWA1**	**AWA2**	**aPY**	**SUN**	**Total**
MV	8	7	7	5	**12**	39
MDT	9	8	7	6	8	38
DNN	7	7	8	**11**	10	43
GT	**10**	**11**	10	**11**	4	**46**
Con	8	10	**12**	8	8	**46**
Auc	9	7	8	6	7	37

The following can be noted on the basis of the results reported in Table 11: (A) *GT* performed the best for the **CUB** and **AWA1** datasets and obtained one of the best overall results alongside *Con* (total score for both equal to 46 points), which achieved the best score for the **AWA2** dataset; (B) *GT* and *DNN* performed the best for the **aPY** dataset; (C) *MV* performed the best for the **SUN** dataset; and (D) *Auc* performed the worst overall (both on individual datasets and for combined performance).

Finally, the same competitive score combination method was applied jointly to both the meta-classifiers and the individual classifiers. For obvious reasons, only the T1 and the F1 accuracy measures were taken into account; since 11 classifiers were compared, the top score was 11 points. The results are displayed in Table 12 (bold font marks the "best" results).

Table 12. Individual Classifier and Meta-Classifier Combined Performance.

CLF	CUB	AWA1	AWA2	aPY	SUN	Total
DeViSE	16	18	18	11	17	80
ALE	**18**	16	16	16	**22**	**88**
SJE	14	13	15	12	18	72
ESZSL	11	9	9	7	8	44
SAE	9	8	11	14	15	57
MV	13	11	13	9	20	66
MDT	16	13	14	14	10	67
DNN	13	12	15	**21**	12	73
GT	16	17	19	**21**	6	79
Con	13	**20**	**21**	17	10	81
Auc	16	12	15	14	9	66

Observations that can be made from the results reported in Table 12 are detailed below:

- For the individual classifiers – *ALE* obtained the highest score on both the **CUB** and **SUN** datasets (18 and 22), as well as the highest overall score (88). Only *ALE* achieved a score of 20 or higher on any given dataset. *DeViSE*, *ALE* and *SJE* obtained overall scores of 60 or higher, whereas *ESZSL* and *SAE* scored below this threshold.
- For the meta-classifiers – *Con* obtained the highest score on both the **AWA1** and **AWA2** datasets (20 and 21); *DNN* and *GT* both obtained the highest score for the **aPY** dataset; *MV*, *DNN*, *GT* and *Con* obtained scores of 20 or higher for any given dataset. All the meta-classifiers achieved an overall score of 60 or higher, with half achieving overall scores of above 70.
- When comparing individual and meta-classifiers, half of the meta-classifiers obtained total scores higher than 70, whereas less than half of the individual classifiers did.
- None of the meta-classifiers obtained total scores of less than 60, while three of the five individual classifiers did.
- Only *ALE* obtained a score of 20 or above on an individual dataset, whereas four of the six (i.e., two-thirds) of the meta-classifiers reached this level.
- Unlike the conclusions presented in [12,13] for the ZSL problem setting, where the simpler meta-classifiers (e.g., *MV*) gave better results, in the case of the GZSL problem, the more complex meta-classifiers (e.g., DNN) delivered better results.

Overall, it can be concluded that, in GZSL settings, selection of the "best approach" is very much context-dependent. With "inside knowledge" of the characteristics of the dataset and/or the aims for obtaining fine-tuned results for a given performance measure, this can be achieved using one of the individual classifiers. On the other hand, when the goal is to solve the problem, while avoiding the "worst case scenario", then, use of meta-classifiers is desirable.

6. Concluding Remarks and Future Research Directions

The aim of this investigation was to experimentally study two different aspects of the generalized zero-shot learning problem. The first involved a comprehensive evaluation of five different approaches (*DeViSE*, *ALE*, *SJE*, *ESZSL* and *SAE*) for solving the GZSL problem, using five standard benchmarking datasets (**CUB**, **AWA1**, **AWA2**, **aPY** and **SUN**), applying four performance measures. The second involved a preliminary assessment of the performance of six standard state-of-the-art meta-classifiers—*MV*, *MDT*, *DNN*, *GT*, *Con* and *Auc*—applied to the results obtained by the five individual GZSL classifiers.

Similar to the conclusions presented in [12,13], at this stage of research, there appears to be no single best overall classifier for the GZSL problem, since the performance of the existing classifiers depends on the dataset and the applied performance measure used. Thus, determining which classifier to use will depend on the data and the goals. For instance, a classifier may be applied when the single best result is important and a different one used when the total performance for the five best results matters.

Another important finding is that the meta-classifiers seem to be valuable as an approach geared towards avoiding performance pitfalls, when it is not clear which solver should be tried. Here, despite their performance ceiling, *MV*, *MDT* and *DNN* performed comparably well. However, the non-constrained models, *GT* and *Con*, achieved the best performances when applied to the GZSL problem.

With respect to future work, first and foremost, it is clear that much more work is needed to develop novel approaches to obtain satisfactory results for the GZSL problem. More work is needed to develop a better understanding of the relationships between the characteristics of the dataset, the method used to solve the GZSL problem, and the performance measure that is being used. There is an obvious need to introduce additional datasets that can be used to gain more knowledge about such relationships. Moreover, studies similar to this one should be undertaken to include methods belonging to other categories of ZSL/GZSL solvers (see Section 3). Finally, similar to the research reported, for instance, in [38], more refined meta-classifiers or meta-learning systems need to be developed and investigated to address the ZSL/GZSL problem. Combining the knowledge obtained following these research projects may result in better understanding of which approaches can raise the quality of available solutions to the next level.

Author Contributions: This contribution is a result of a team effort, supported by multiple funding and cooperation agreements. While the lead author was responsible for coding and running the experiments, the remaining authors participated equally in all remaining aspects of the project. All authors have read and agreed to the published version of the manuscript.

Funding: This work is funded in part by the Centre for Priority Research Area Artificial Intelligence and Robotics of the Warsaw University of Technology within the Excellence Initiative: Research University (IDUB) programme. Moreover it was supported by and realized within the framework of a bilateral project between the Polish Academy of Sciences and the Romania Academy, entitled "Semantic foundation of the Internet of Things", a bilateral project IC-PL/01/2022–2023 between the Polish Academy of Sciences and the Bulgarian Academy of Sciences, entitled "Practical aspects of scientific computing", as well as a collaboration agreement between the University of Novi Sad, University of Craiova, SRI PAS and the Warsaw University of Technology.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

ZSL	Zero-shot learning
GZSL	Generalized zero-shot learning
DeViSE	Deep visual semantic embedding
ALE	Attribute label embedding
SJE	Structured joint embedding
ESZSL	Embarrassingly simple approach to zero-shot learning
SAE	Semantic autoencoder
MDT	Meta-decision tree
DNN	Deep neural network
GT	Game-theory-based approach
Auc	Auction-based model
Con	Consensus-based approach
T1	Top-1
T5	Top-5

LogLoss	Logarithmic loss
F1	F1 score
CUB	Caltech-UCSD-Birds 200-2011
SUN	Scene UNderstanding
AWA1	Animals with Attributes 1
AWA2	Animals with Attributes 2
aPY	Attribute Apascal&Ayahoo

References

1. Chao, W.L.; Changpinyo, S.; Gong, B.; Sha, F. An empirical study and analysis of generalized zero-shot learning for object recognition in the wild. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016; Springer: Berlin/Heidelberg, Germany, 2016; pp. 52–68.
2. Joachims, T. Transductive learning via spectral graph partitioning. In Proceedings of the 20th International Conference on Machine Learning (ICML-03), Washington, DC, USA, 21–24 August 2003; pp. 290–297.
3. Arnold, A.; Nallapati, R.; Cohen, W.W. A comparative study of methods for transductive transfer learning. In Proceedings of the Seventh IEEE international conference on data mining workshops (ICDMW 2007), Omaha, NE, USA, 28–31 October 2007; IEEE: Piscataway, NJ, USA, 2007; pp. 77–82.
4. Palatucci, M.M. *Thought Recognition: Predicting and Decoding Brain Activity Using the Zero-Shot Learning Model*; Carnegie Mellon University: Pittsburgh, PA, USA, 2011.
5. Lampert, C.; Nickisch, H.; Harmeling, S. Attribute-based classification for zero-shot learning of object categories. *IEEE Trans. Pattern Anal. Mach. Intell.* **2013**, *36*, 453–465. [CrossRef] [PubMed]
6. Larochelle, H.; Erhan, D.; Bengio, Y. Zero-data learning of new tasks. In Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence, AAAI 2008, Chicago, IL, USA, 13–17 July 2008; Volume 1, p. 3.
7. Rohrbach, M.; Stark, M.; Schiele, B. Evaluating knowledge transfer and zero-shot learning in a large-scale setting. In Proceedings of the CVPR 2011, Springs, CO, USA, 20–25 June 2011; IEEE: Piscataway, NJ, USA, 2011; pp. 1641–1648.
8. Yu, X.; Aloimonos, Y. Attribute-based transfer learning for object categorization with zero/one training example. In Proceedings of the European Conference on Computer Vision, Crete, Greece, 5–11 September 2010; Springer: Berlin/Heidelberg, Germany, 2010; pp. 127–140.
9. Xu, X.; Shen, F.; Yang, Y.; Zhang, D.; Tao Shen, H.; Song, J. Matrix tri-factorization with manifold regularizations for zero-shot learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 3798–3807.
10. Ding, Z.; Shao, M.; Fu, Y. Low-rank embedded ensemble semantic dictionary for zero-shot learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2050–2058.
11. Xian, Y.; Lampert, C.H.; Schiele, B.; Akata, Z. Zero-shot learning—A comprehensive evaluation of the good, the bad and the ugly. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *41*, 2251–2265. [CrossRef] [PubMed]
12. Saad, E.; Paprzycki, M.; Ganzha, M. Practical Aspects of Zero-Shot Learning. In Proceedings of the Computational Science—ICCS 2022, London, UK, 21–23 June 2022; Groen, D., de Mulatier, C., Paszynski, M., Krzhizhanovskaya, V.V., Dongarra, J.J., Sloot, P.M.A., Eds.; Springer International Publishing: Cham, Switzerland, 2022; pp. 88–95.
13. Saad, E.; Paprzycki, M.; Ganzha, M. Practical Aspects of Zero-Shot Learning. *arXiv* **2022**, arXiv:2203.15158.
14. Jain, S.; Kotsampasakou, E.; Ecker, G.F. Comparing the performance of meta-classifiers—a case study on selected imbalanced data sets relevant for prediction of liver toxicity. *J. Comput.-Aided Mol. Des.* **2018**, *32*, 583–590. [CrossRef] [PubMed]
15. Frome, A.; Corrado, G.S.; Shlens, J.; Bengio, S.; Dean, J.; Ranzato, M.; Mikolov, T. Devise: A deep visual-semantic embedding model. *Adv. Neural Inf. Process. Syst.* **2013**, *26*, 2121–2129.
16. Akata, Z.; Perronnin, F.; Harchaoui, Z.; Schmid, C. Label-embedding for image classification. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *38*, 1425–1438. [CrossRef] [PubMed]
17. Akata, Z.; Reed, S.; Walter, D.; Lee, H.; Schiele, B. Evaluation of output embeddings for fine-grained image classification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 2927–2936.
18. Shalev-Shwartz, S.; Ben-David, S. *Understanding Machine Learning: From Theory to Algorithms*; Cambridge University Press: Cambridge, UK, 2014.
19. Romera-Paredes, B.; Torr, P. An embarrassingly simple approach to zero-shot learning. In Proceedings of the International Conference on Machine Learning, PMLR, Lille, France, 7–9 July 2015; pp. 2152–2161.
20. Xian, Y.; Akata, Z.; Sharma, G.; Nguyen, Q.; Hein, M.; Schiele, B. Latent embeddings for zero-shot classification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 69–77.
21. Socher, R.; Ganjoo, M.; Sridhar, H.; Bastani, O.; Manning, C.D.; Ng, A.Y. Zero-shot learning through cross-modal transfer. *arXiv* **2013**, arXiv:1301.3666.
22. Zhang, Z.; Saligrama, V. Zero-shot learning via semantic similarity embedding. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 4166–4174.
23. Norouzi, M.; Mikolov, T.; Bengio, S.; Singer, Y.; Shlens, J.; Frome, A.; Corrado, G.S.; Dean, J. Zero-shot learning by convex combination of semantic embeddings. *arXiv* **2013**, arXiv:1312.5650.

24. Changpinyo, S.; Chao, W.L.; Gong, B.; Sha, F. Synthesized classifiers for zero-shot learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 5327–5336.
25. Verma, V.K.; Rai, P. A simple exponential family framework for zero-shot learning. In Proceedings of the Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Skopje, Macedonia, 18–22 September 2017; Springer: Berlin/Heidelberg, Germany, 2017; pp. 792–808.
26. Ye, M.; Guo, Y. Zero-shot classification with discriminative semantic representation learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 7140–7148.
27. Todorovski, L.; Džeroski, S. Combining classifiers with meta decision trees. *Mach. Learn.* **2003**, *50*, 223–249. [CrossRef]
28. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; MIT Press: Cambridge, MA, USA, 2016.
29. Abreu, M.d.C.; Canuto, A.M. Analyzing the benefits of using a fuzzy-neuro model in the accuracy of the neurage system: An agent-based system for classification tasks. In Proceedings of the 2006 IEEE International Joint Conference on Neural Network, Vancouver, BC, Canada, 16–21 July 2006; IEEE: Piscataway, NJ, USA, 2006; pp. 2959–2966.
30. Alzubi, O.A.; Alzubi, J.A.A.; Tedmori, S.; Rashaideh, H.; Almomani, O. Consensus-based combining method for classifier ensembles. *Int. Arab J. Inf. Technol.* **2018**, *15*, 76–86.
31. Ruta, D.; Gabrys, B. Classifier selection for majority voting. *Inf. Fusion* **2005**, *6*, 63–81. [CrossRef]
32. Sokolova, M.; Japkowicz, N.; Szpakowicz, S. Beyond accuracy, F-score and ROC: A family of discriminant measures for performance evaluation. In Proceedings of the Australasian Joint Conference on Artificial Intelligence, Hobart, Australia, 4–8 December 2006; Springer: Berlin/Heidelberg, Germany, 2006.
33. Ferger, W.F. The nature and use of the harmonic mean. *J. Am. Stat. Assoc.* **1931**, *26*, 36–40. [CrossRef]
34. Nwankpa, C.; Ijomah, W.; Gachagan, A.; Marshall, S. Activation functions: Comparison of trends in practice and research for deep learning. *arXiv* **2018**, arXiv:1811.03378.
35. Welinder, P.; Branson, S.; Mita, T.; Wah, C.; Schroff, F.; Belongie, S.; Perona, P. *Caltech-UCSD Birds 200*; Technical Report 2010-001; California Institute of Technology: Pasadena, CA, USA, 2010.
36. Patterson, G.; Hays, J. Sun attribute database: Discovering, annotating, and recognizing scene attributes. In Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, 16–21 June 2012; IEEE: Piscataway, NJ, USA, 2012; pp. 2751–2758.
37. Farhadi, A.; Endres, I.; Hoiem, D.; Forsyth, D. Describing objects by their attributes. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; IEEE: Piscataway, NJ, USA, 2009; pp. 1778–1785.
38. Demertzis, K.; Iliadis, L. GeoAI: A model-agnostic meta-ensemble zero-shot learning method for hyperspectral image analysis and classification. *Algorithms* **2020**, *13*, 61. [CrossRef]