POLITECNICO DI TORINO
Repository ISTITUZIONALE

Fuzzy Logic Visual Network (FLVN): A neuro-symbolic approach for visual features matching

Original
Fuzzy Logic Visual Network (FLVN): A neuro-symbolic approach for visual features matching / Manigrasso, F., Morra, L., Lamberti, F.. - ELETTRONICO. - 14234:(2023), pp. 456-467. (22nd International Conference on IMAGE ANALYSIS AND PROCESSING (ICIAP 2023) Udine 11-15 settembre 2023) [10.1007/978-3-031-43153-1_38].

Availability:
This version is available at: 11583/2980349 since: 2023-09-21T10:01:21Z

Publisher:
Springer

Published
DOI:10.1007/978-3-031-43153-1_38

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright
Springer postprint/Author's Accepted Manuscript

This version of the article has been accepted for publication, after peer review (when applicable) and is subject to Springer Nature's AM terms of use, but is not the Version of Record and does not reflect post-acceptance improvements, or any corrections. The Version of Record is available online at: http://dx.doi.org/10.1007/978-3-031-43153-1_38

(Article begins on next page)

08 November 2026

Fuzzy Logic Visual Network (FLVN): A neuro-symbolic approach for visual features matching

Francesco Manigrasso[1 [0000−0002−4151−8880]], Lia Morra[1 [0000−0003−2122−7178]], and Fabrizio Lamberti[1 [0000−0001−7703−1372]]

Politecnico di Torino, Dipartimento di Automatica e Informatica, Torino, Italy
{francesco.manigrasso, lia.morra, fabrizio.lamberti}@polito.it

Abstract. Neuro-symbolic integration aims at harnessing the power of symbolic knowledge representation combined with the learning capabilities of deep neural networks. In particular, Logic Tensor Networks (LTNs) allow to incorporate background knowledge in the form of logical axioms by grounding a first order logic language as differentiable operations between real tensors. Yet, few studies have investigated the potential benefits of this approach to improve zero-shot learning (ZSL) classification. In this study, we present the Fuzzy Logic Visual Network (FLVN) that formulates the task of learning a visual-semantic embedding space within a neuro-symbolic LTN framework. FLVN incorporates prior knowledge in the form of class hierarchies (classes and macro-classes) along with robust high-level inductive biases. The latter allow, for instance, to handle exceptions in class-level attributes, and to enforce similarity between images of the same class, preventing premature overfitting to seen classes and improving overall performance. FLVN reaches state of the art performance on the Generalized ZSL (GZSL) benchmarks AWA2 and CUB, improving by 1.3% and 3%, respectively. Overall, it achieves competitive performance to recent ZSL methods with less computational overhead. FLVN is available at `https://gitlab.com/grains2/flvn`.

Keywords: Zero shot learning · NeuroSymbolic AI · Logic Tensor Networks

1 Introduction

There is an increasing interest in the computer vision community towards the integration of learning and reasoning, with specific emphasis towards the fusion of deep neural networks and symbolic artificial intelligence [1–3]. By integrating perception, reasoning and learning, Neuro-symbolic (NeSy) architectures can improve explainability, generalization and robustness of deep learning systems. In addition, NeSy techniques can incorporate prior symbolic knowledge in the training objective of deep neural networks, compensating for the lack of supervision from labelled examples [4, 5]. The latter property is of particular interest in the context of Zero-Shot Learning (ZSL), which aims at recognizing objects

from unseen classes by associating both seen and unseen classes to auxiliary information, usually in the form of class attributes [6].

In this paper, we introduce the Fuzzy Logic Visual Network (FLVN), an architecture designed to learn a joint embedding space for visual features and class attributes to tackle ZSL scenarios. To constraint learning based on prior knowledge available from resources such as WordNet, it leverages the Logic Tensor Network (LTN)[2] paradigm, a NeSy framework that formulates learning as maximizing the satisfiability of a knowledge base $\mathcal{K}$ based on a first order logic (FOL). Thus, FLVN is composed by a convolutional neural network (CNN), that computes embedding, followed by a LTN.

The proposed FLVN improves and extends previous NeSy architectures for ZSL, and in particular our previous architecture ProtoLTN [7], in several ways. First, there are substantial differences in the LTN formulation, and especially the way the isOfClass predicate is formulated. FLVN, learns to project visual features into the semantic attribute space, facilitating the integration of prior knowledge into the learning process by the LTN. FLVN is trained end-to-end, whereas ProtoLTN used features extracted from a pre-trained frozen network, and only the class-level prototypes were trained. This implementation outperforms our previous model [7], offering an alternative approach for grounding the predicates in the FOL language by changing the how the distance measure between extracted features and semantic vectors is calculated.

Compared to most existing approaches to ZSL, FLVN incorporates into the training process several logical axioms that combine both class-specific prior knowledge (e.g., "a zebra is an ungulate") with high-level inductive biases (e.g., "two images of the same class should have similar embeddings"), that were not yet included in ProtoLTN [7]. Finally, we introduce axioms to represent exceptions within the dataset to make the architecture more robust in the context of ZSL, where class attributes are specified at the class level rather than the image level. To the best of our knowledge, this aspect is not taken into account in current approaches to ZSL, including neuro-symbolic ones [7]. Through extensive experiments on multiple benchmarks, we show that FLVN outperforms existing ZSL architectures. The implementation, developed in PyTorch using the LTNtorch package [8], is available at `https://gitlab.com/grains2/flvn`

The remainder of the paper is structured as follows. Section 2 introduces background on the LTN framework and related work. Section 3 describes the proposed architecture. In Sections 4 and 5, we analyze the model behavior in a ZSL and a generalized ZSL (GZSL) setting on common benchmarks. Finally, in Section 6, we discuss the results and future work.

2 Related work

2.1 Neural-symbolic AI in semantic image interpretation

In recent years, there has been significant research focus on NeSy architectures for addressing semantic image interpretation tasks [1, 2, 4, 5, 7, 9]. The present

studies falls within the class of NeSy techniques that seeks to incorporate symbolic information as a prior [3]. Specifically, we rely on LTNs, which are modular architectures capable of incorporating FOL constraints [2, 7] and can be jointly trained with neural module in an end-to-end manner [4].

In the LTN framework, the concept of *grounding* plays a crucial role in interpreting FOL within a specific subset of the domain, denoted as $\mathbb{R}^n$. In this approach, logical predicates and axioms are represented as vectors, which are then grounded (interpreted) as real numbers in the range [0, 1] using a technique called *Real Logic*. By employing this grounding mechanism, the LTN framework maps each term, denoted as x, to a vector representation in $\mathcal{G}(x) = \mathbb{R}^n$, while each predicate symbol, represented by $p \in \mathcal{P}$, is mapped to $\mathcal{G}\left(D(p)\right) \rightarrow [0, 1]$. The concept can be illustrated by considering the frequently encountered `isOfClass` predicate in neuro-symbolic architectures, which quantifies the probability of a given term belonging to a specific class c [4, 7]. The training objective is formulated by constructing a knowledge base $\mathcal{K}$ of FOL axioms, and finding the *best satisfiability* (sat).

2.2 Zero-shot learning

ZSL tasks entail recognizing objects from previously unseen classes by exploiting some form of auxiliary knowledge, usually attribute-based, learned from seen classes. GZSL extends ZSL by assuming both seen and unseen classes are present at test time. Different strategies have been proposed to tackle ZSL, including embedding-based, attention mechanism-based, and generative strategies.

Embedding-based techniques compare semantic attributes and visual information by mapping them onto a suitable embedding space. Some methods embed images into the attribute space using an embedding function and consider semantic attributes as the common space [10, 11]. Other techniques have proposed to adopt the image embedding space as common space [7, 12, 13] to mitigate the hubness problem [14]. A third class of approaches [15–17] utilizes a shared space distinct from both image and attribute domains. To prevent overfitting to seen classes, these approaches require the use of pseudo-labeling techniques or a transductive setting [16], assuming that unlabeled images from unseen classes are provided during training. **Attention mechanisms** have been proposed to find the image regions that contribute the most to the categorization of a certain class and improve the embedding space [10, 11, 18, 19].

A critical aspect of embedding-based method is preventing overfitting to seen classes using, e.g., regularization [10, 16] or contrastive techniques [11]. FLVN accomplishes this objective by incorporating a symbolic prior to aggregate visually and semantically similar features, while also explicitly establishing relationships between classes within the same macro class (a characteristic often implicitly addressed in alternative methodologies).

Previous approaches have used semantic class descriptions for the classification, but attributes linked to a specific class may not be consistently expressed or detectable in individual images. FLVN addresses this limitation by incorporating axioms that encode the existence of "exceptions to the rule" within the dataset.

This aspect, which has received limited consideration in previous studies, has been experimentally shown to improve classification accuracy.

Generative techniques, on othe other hand, exploit auxiliary models, such as Generative Adversarial Networks (GANs), to generate artificial examples representing unseen classes by learning a conditional probability for each class [20–23]. Recently, feature generation models were integrated with embedding-based models in a contrastive setting [24]. Generative methods require prior knowledge regarding unseen classes when generating training data. In contrast, the training process of FLVN relies on a subset of distinct seen classes and solely assumes knowledge of the class hierarchy at training time, so that the training can be easily extended to new unseen classes. Nonetheless, our approach is complementary to generative techniques, and could be in principle combined.

3 Fuzzy Logic Visual Network

We introduce a comprehensive and trainable framework for the task of ZSL, depicted in Figure 1. It comprises two main modules: a feature extractor and a LTN that formulates the training objective.

Feature extractor. The feature (embedding) extractor is a CNN that maps the input x to a feature space $f_\theta(x) \in \mathbb{R}^{H\times W\times B}$, where H, W and B represent the height, width, and number of channels of the features, respectively. Through mean pooling over H and W, we obtain global discriminative characteristics $g_\theta(x) \in \mathbb{R}^{B\times 1}$, and utilize a linear projection to transform them into a semantic space represented by $V \in \mathbb{R}^{B\times M}$, where B is the dimension of the vector space of features, and M denotes the length of the attribute vector [10].

Logic Tensor Network. The LTN formulates the learning objective as the maximum satisfiability of a $\mathcal{K}$. For each training batch, the $\mathcal{K}$ is updated introducing axioms that represent labelled examples (ϕ_1), as well as prior knowledge ($\phi_2, \phi_3, \phi_4, \phi_5, \phi_6$). The maximum satisfiability loss is then defined based on the aggregation of all axioms as follows:

$$\mathcal{L}^{\text{ep}} = 1 - \left(\bigwedge_{\phi\in\mathcal{K}} \phi \right) = 1 - \mathcal{G}(\phi) \tag{1}$$

This section first define the variables, predicates and domain that form the FOL language, followed by the definition of the knowledge base $\mathcal{K}$.

Groundings. Variables and their domains are grounded as follows:

$$\mathcal{G}(l) = \mathbb{N}^C, \mathcal{G}(q) = \mathbb{N}^Q \tag{2}$$

$$\mathcal{G}(a) = \mathcal{G}(a^{\text{mask}}) = \mathbb{R}^{M\times C} \tag{3}$$

$$\mathcal{G}(a^{\text{macro}}) = \mathbb{R}^{M\times Q} \tag{4}$$

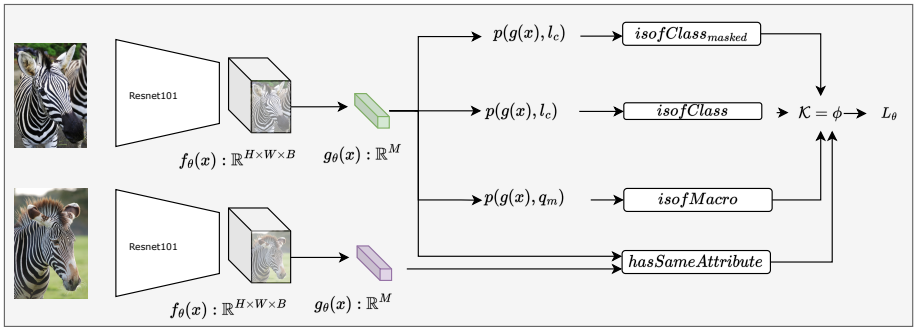

Fig. 1. The FLVN architecture, designed for Zero-Shot Learning (ZSL) classification, combines a convolutional feature extractor and an attribute encoder to effectively map visual features to the attribute space. By leveraging predicates such as `isOfClass`, $\texttt{isOfClass}_{\text{masked}}$, and `isOfMacro`, it reduces the disparity between image features and class attributes, while the `hasSameAttribute` predicate captures the similarity between two images. All these predicates play a crucial role in the formulation of the architecture, as they are incorporated within the formulas of a knowledge base denoted as $\mathcal{K}$. During the training process, the loss function is designed to optimize the satisfiability (truth value) of all the formulas within $\mathcal{K}$.

$$\mathcal{G}(x) = g_\theta(f_\theta(\mathcal{G}(\texttt{images}))) = \mathbb{R}^M \tag{5}$$

where the variable l represents the class labels belonging to set of classes C, q represent the macroclass label belonging to set of macroclasses Q, and each class/macroclass is described by a set of non-binary semantic attributes denoted by a and a^{macro}, respectively. Functions f_θ and g_θ are employed to embed images into the attribute space, resulting in the final representation $\mathcal{G}(x)$. The FOL language contains four main predicates: $\texttt{isOfClass}(x, l)$ and $\texttt{isOfClass}_{\text{masked}}(x, l)$ denote the fact that an image x belongs to class l, $\texttt{isOfMacro}(x, q)$ that an image x belongs to the macroclass q, and $\texttt{hasSameAttribute}(x_1, x_2)$ that two images have the same attributes.

The $\mathcal{G}(\texttt{isOfClass})$ predicate is grounded by the similarity between the input image and the corresponding class attribute vectors. First, we compute the similarity between the image x and a class l_c by calculating the scaled product of the global features mapped in the attribute space with the semantic vectors:

$$p\left(x, l\right) = \frac{\exp\left(x^T V a_l\right)}{\sum_{s=1}^{S} \exp\left(x^T V a_s\right)} \tag{6}$$

where a_y represents the semantic attribute vector associated with class l. To obtain a prediction score for an example x, we calculate the dot product between the output of p (Eq. 6) and the one-hot encoding l_c^T for class $c \in C$, as follows:

$$\mathcal{G}(\texttt{isOfClass}) : x, l_c \rightarrow {l_c}^T p(\mathcal{G}(x), l_c) \tag{7}$$

Similarly, we define $\mathcal{G}(\texttt{isOfMacro})$ and $\mathcal{G}(\texttt{isOfClass}_{\text{masked}})$. Since attributes for macro-classes are in principle unknown, we define a trainable attribute vector a_m^{macro} for macro-class q_m to compute $\mathcal{G}(\texttt{isOfMacro})$. On the other hand, $\mathcal{G}(\texttt{isOfClass}_{\text{masked}})$ uses the masked attribute vector a_c^{masked} for class l_c, in which missing attributes k are set to 0, while preserving the rest of the attributes in a_c. Finally, the grounding for `hasSameAttribute` is defined as:

$$\mathcal{G}(\texttt{hasSameAttribute}) : x_1, x_2 \rightarrow \texttt{sigmoid}(\alpha \texttt{d}(\mathcal{G}(x_1), \mathcal{G}(x_2))) \tag{8}$$

where `d` is the cosine similarity, α a scale factor and $\mathcal{G}(x_1)$, $\mathcal{G}(x_2)$ correspond to the embeddings of the two images.

Learning from labeled examples. We incorporate labelled examples by introducing an axiom ϕ_1 stating that all facts about labeled example should be true, that is, all labeled samples should be classified correctly:

$$\phi_1 = \forall \text{Diag}(x, l_c)(\texttt{isOfClass}(x, l_c))^1 \tag{9}$$

To account for the class hierarchy, we also introduce an axiomatic statement ϕ_2 to indicate that "if an image contains a zebra", then "the image belongs to the family of ungulates":

$$\phi_2 = \forall \text{Diag}(x, l_c, q_m)(\texttt{isOfClass}(x, l_c) \implies \texttt{isOfMacro}(x, q_m)) \tag{10}$$

Learning better feature representations. The following axiom encodes the assumption that features extracted from two images of the same class should possess the same attributes:

$$\phi_3 = \forall \text{Diag}(x_1, l_{c_1}) \Big(\forall \text{Diag}(x_2, l_{c_2}) : c_1 = c_2 \ \texttt{hasSameAttribute}(x_1, x_2) \Big) \tag{11}$$

Likewise, images from different classes should possess different attributes:

$$\phi_4 = \forall \text{Diag}(x_1, l_{c_1}) \Big(\forall \text{Diag}(x_2, l_{c_2}) : c_1 != c_2 \ \neg \texttt{hasSameAttribute}(x_1, x_2) \Big) \tag{12}$$

To further emphasize the similarity between visual attributes and semantic vectors, the following axiom enforces the similarity between image embeddings and attribute vectors of the same class:

$$\phi_5 = \forall \text{Diag}(x, l_c) \Big(\forall \text{Diag}(a, l_a) : c = a \ \texttt{hasSameAttribute}(x, a) \Big) \tag{13}$$

Learning with refutation. In classical ZSL benchmarks such as AWA2, attributes are associated with class labels using a crisp or fuzzy matrix. However, this association does not entail that all examples of a class will exhibit exactly

[1] Diagonal Quantification quantifies over pairs of instances, e.g., images and their labels. A more formal definition can be found in [2].

the same attributes: attributes may be expressed by a subset of training samples, or may be occluded. The existential statement ϕ_6 represents the fact that some class attributes may not be present for all samples (e.g, "there exists a zebra that is not agile"). Given that image-level attributes are not available, we simply remove randomly selected attributes by defining the $\texttt{isOfClass}_{\text{masked}}$ predicate:

$$\phi_6 = \forall l^{\text{seen}}(\exists x, \texttt{isOfClass}_{\text{masked}}(x, l^{\text{seen}})) \tag{14}$$

where l^{seen} denotes the list of seen classes.

Grounding logical connectives and aggregators. The knowledge base $\mathcal{K}$ is an aggregation of formulas updated at each training step. To solve the maximum satisfiability problem using gradient descent, logical connectives and aggregators must be grounded into Real Logic. Given two truth values a and b in $[0, 1]$, we adopted the symmetric configuration from [2], using the standard negation $\neg : N_S(a) = 1 - a$ and the Reichenbach implication $\rightarrow$: $I_R(a, b) = 1 - a + ab$. The existential quantifier $\exists$ was approximated by the generalized mean A_{pM}, and the universal quantifier $\forall$ by the generalized mean w.r.t. the error A_{pME}, respectively [2, 25]. Given n truth values $a_1, \ldots, a_n$ all in $[0, 1]$:

$$\exists : A_{pM}\left(a_1, \ldots, a_n\right) = \left(\frac{1}{n}\sum_{i=1}^{n} a_i^{p_\exists}\right)^{\frac{1}{p_\exists}} \quad p_\exists \geqslant 1 \tag{15}$$

$$\forall : A_{pME}\left(a_1, \ldots, a_n\right) = 1 - \left(\frac{1}{n}\sum_{i=1}^{n} \left(1 - a_i\right)^{p_\forall}\right)^{\frac{1}{p_\forall}} \quad p_\forall \geqslant 1 \tag{16}$$

A_{pME} is a measure of how much, on average, truth values a_i deviate from the true value of 1. The A_{pME} was also used to approximate $\bigwedge$ in Eq.[1]. Further details on the role of $p_\exists$ and $p_\forall$ can be found in previous works [2].

Querying the knowledge base. At inference time, the class with the highest score is selected as the predicted class:

$$\hat{y} = \underset{\tilde{y} \in Y^U}{argmax}(g(x)^{\mathrm{T}} V a_{\tilde{y}}) \tag{17}$$

FVLN was evaluated in both ZSL and GZSL settings. In the ZSL setting, only unseen images are assumed to be present at test time, whereas in the GZSL setting, the model is tested on both seen and unseen classes. This setup induces a bias towards seen classes. To mitigate it, we employed the Calibrated Stacking method, as proposed in [26, 27], to diminish the classification score of seen classes. The class score is thus calculated as $\hat{y}$:

$$\hat{y} = \underset{\tilde{y} \in Y^U \cup Y^S}{argmax}\left(g(x)^{\mathrm{T}} V a_{\tilde{y}} - \gamma \mathbb{I}\left[\tilde{y} \in \mathcal{Y}^S\right]\right) \tag{18}$$

where $\mathbb{I} = 1$ if $\tilde{y}$ is from a seen class and zero otherwise, γ is a calibration coefficient tuned on a validation set and $\mathcal{Y}^S$ are the labels of seen classes.

Construction of the training batch. Following the approach in [11], for a positive input image x_i, we select a set of positive examples x^+ and K negative examples $x_1^-, ..., x_K^-$. Positive examples are selected from the same category as x_i, while negative examples are randomly selected from the remaining classes.

4 Experimental settings

In this section we examine the datasets used in our experiments, the knowledge base adopted for each dataset and the list of hyperparameters chosen.

Dataset. The experiments were conducted on the AWA2 [6], CUB [28], and SUN [29] benchmarks. The evaluation metrics for GZSL were based on the standards defined in previous work [6]. Following a similar approach outlined in [30], we construct a semantic hierarchy by grouping the classes from Awa2 and CUB datasets into a total of 9 and 49 macroclasses, respectively.

Knowledge base. The knowledge base $\mathcal{K}$ composition was detailed in Section 3. To build class hierarchy, we map classes from the AWA2 and CUB datasets to their corresponding synsets in WordNet as done by Sikka et al. [30]. Macroclasses for each dataset were defined by selecting the synset root whose subtree contained classes from the selected dataset, and then defining its immediate children (using WordNet) as classes. Since classes in the SUN dataset [29] lack a semantic hierarchy, axioms related to macroclasses were not included in the knowledge base. For all experiments, in the predicate $\texttt{isofClass}_{masked}$, $k = 15$ attributes were randomly dropped. The α parameter in Eq. 8 was set to 0.01 for AWA2 dataset and 1 on CUB e SUN. To account for the presence of outliers in the knowledge base, we initially set the aggregation function parameters (defined in Eqs. 15 and 16) as $p_{\exists} = 2$ and $p_{\forall} = 2$. Both parameters were incremented by 2 every 4 epochs for the AWA2 and CUB datasets; for SUN, the aggregation values were increased at specific epochs (2, 4, 24, and 32) until reaching $p_{\exists} = 6$ and $p_{\forall} = 6$, following the schedule suggested in [2].

Hyperparameter selection. The embedding function f_θ is based on an ImageNet pre-trained ResNet101 model that converts the 224×224 image into a vector $\mathbf{x} \in \mathbb{R}^{H \times W \times B}$, where $B = 2048$, and H and W represent the height and width of the extracted features; the function g_θ then converts these features into the attribute space dimensionally consistent with the dataset.

To mitigate overfitting, we initially trained the head with a frozen backbone and subsequently fine-tuned the entire network. We used the Adam optimizer with learning rate of $1e-4$ for the pre-training phase for AWA2 and CUB and 5e-4 for SUN. The learning rate was then reduced to $1e-6\alpha$ for CUB and SUN and $1e-7\alpha$ for AWA2 in the fine-tuning phase, where $\alpha = 0.8^{epoch//10}$ and *epoch* is the current epoch out of a total of 300 epochs. Additionally, we incorporated a smoothing factor by multiplying the L2 norm with a scaling factor of 5e-4 for AWA2, 5e-6 for CUB, and 1e-3 for SUN.

Training batches included positive and negative examples with a ratio of 12 to 12 for AWA2, 12 to 8 for CUB, and 12 to 4 for SUN datasets. At inference time, we set the scaling factor γ to adjust the scores obtained for the seen classes to 0.7 for AWA2 and CUB and 0.4 for SUN. In all experiments, we applied a random crop and random flip with 0.5 probability for data augmentation. We implemented the architecture in PyTorch, using the LTNtorch library [**?**], and trained it on a single GPU nVidia 2080 Ti. Each experiment was repeated three times to calculate the mean and standard deviation.

	AWA2				CUB				SUN			
Model	T1	U	S	H	T1	U	S	H	T1	U	S	H
PROTO-LTN [7]	67.6	32.0	83.7	46.2	48.8	20.8	54.3	30.0	60.4	20.4	**36.8**	26.2
DEM [12]	67.1	30.5	**86.4**	45.1	51.7	19.6	57.9	29.2	61.9	20.5	34.3	25.6
VSE [13]	84.4	45.6	88.7	60.2	71.9	39.5	68.9	50.2	-	-	-	-
TCN [17]	71.2	61.2	65.8	63.4	59.5	52.6	52.0	52.3	61.5	31.2	37.3	34.0
CSNL [30] †	61.0	-	-	0.0	32.5	-	-	0.7	-	-	-	-
AREN [19]	67.9	54.7	79.1	64.7	71.8	63.2	69.0	66.0	60.6	40.3	32.3	35.9
APN [10]	68.4	56.5	78.0	65.5	72.0	65.3	69.3	67.2	61.6	41.9	34.0	37.6
AMGML [18]	71.7	56.0	74.6	64.0	70.0	58.2	55.7	56.9	59.7	42.0	35.1	38.3
CC-ZSL[11]	68.8	62.2	83.1	71.1	74.3	66.1	73.2	69.5	62.4	44.4	36.9	40.3
Cycle-CLSWGAN [23]	-	-	-	-	58.4	45.7	61.0	52.3	60.0	**49.4**	33.6	40.0
LisGan [22]	-	-	-	-	58.8	46.5	57.9	51.6	60.0	42.9	37.8	40.2
E-PGN [20]	73.4	52.6	83.5	64.6	72.4	52.0	61.1	56.2	-	-	-	-
TGMZ [21]	**78.4**	64.1	77.3	70.1	66.1	60.3	56.8	58.5	-	-	-	-
CEGZSL [24]	70.4	63.1	78.6	70.0	77.5	63.9	66.8	65.3	63.3	48.8	38.6	43.1
DFCA-GZSL [31]	74.7	66.5	81.5	73.3	**80.0**	**70.9**	63.1	66.8	62.6	48.9	**38.8**	**43.3**
FLVN †	69.8	65.8	82.3	73.1	71.2	62.6	83.1	71.5	61.7	48.4	32.7	39.0
	±0.8	±0.9	±0.3	±0.6	±0.2	±0.5	±0.3	±0.2	±0.18	±0.57	±0.2	±0.1
	(71.0)	(67.1)	(82.8)	(**74.1**)	(71.4)	(63.2)	(83.4)	(**71.7**)	(61.9)	(48.9)	(32.9)	(39.1)

Table 1. Performance on AWA2, CUB and SUN test set. For FLVN, we show mean ± standard deviation and maximum (in parentheses) values for TOP1zsl (T1), TOP1gzsl unseen (U), TOP1gzsl seen (S), and Hgzsl (H) over three runs. Metrics are described in [6]. The first section of the table includes embedding-based models, the second section attention-based models, and the third section generative models. The best performance values are shown in bold. † highlights method which incorporate external knowledge.

5 Results

Experimental results presented in Table 1 show how the proposed FLVN architecture reaches competitive performances with respect to other embedding-based methods, particularly CC-ZSL [11] and APN [10]. In general, FVLN performs better in terms of harmonic mean (H) and shows a greater ability to recognize both seen and unseen classes, achieving state-of-the-art performance on two out of three benchmark datasets. In particular, FLVN improves the accuracy of unseen classes and the harmonic mean by 0. 89% and 1.3%, respectively, in AWA2, and the accuracy of seen classes and harmonic mean by 12% and 3%, respectively, on CUB. These results suggest that the proposed architecture is capable of recognizing seen and unseen samples more evenly, showing less confusion in classifying examples. On SUN, FLVN performs comparably to the state of the art; it is important to note that, in this case, axioms on macro classes are not introduced, therefore FLVN cannot leverage external semantic knowledge.

Compared to the closest methods in terms of performance, the proposed method requires a relatively simple architecture. Unlike APN, FVLN does not require additional weights for each attribute or supplemental regularization terms to construct the loss function. FVLN only requires a single backbone replica instead of two as in CC-ZSL, which is based on the teacher-student framework [11].

Compared to generative methods, FLVN does not require to generate additional training samples or make assumptions about unseen classes at train time.

Although complete ablation studies are left to future work, qualitative observations from our experiments suggest that axioms included in the knowledge base may have quite different impact on the overall performance. With respect to our previous architecture [7], it was crucial to change the `isOfClass` grounding for end-to-end training to succeed. The introduction of the `isOfMacroclass` predicate, on the other hand, introduces smaller improvements. In fact, during training related classes appear to naturally cluster in feature space, as evident from the distribution of the feature space already reported in previous work [7]. On the other hand, the `isOfClass_masked` offered important benefits, allowing to account for mistakes in the semantic annotation of classes or traits that exist at the class level but are not visible or easily inferred from a single image (e.g., "All zebras are agile"). Finally, the `hasSameAttribute` predicate improved performance particularly on the CUB (fine-grained bird recognition) and SUN (scene recognition) datasets, in which extracting fine-grained image-level attributes is more difficult.

6 Conclusions and future work

Building upon principles from the recent ZSL literature [10] and incorporating them within a NeSy framework [7], we introduced a novel NeSy architecture, named FLVN, for ZSL and GZSL tasks. FLVN incorporates axioms that combine prior class-specific knowledge (e.g., class hierarchies), with high-level inductive biases to handle exceptions within the dataset (e.g., "there exists a zebra that is not agile") and establish relationships between images (e.g., "if two images belong to the same class, they must be similar"). Such axioms act as semantic priors and compensate for the lack of annotations, thereby providing a solid NeSy foundation for GZSL tasks. FLVN does not require multiple backbones and maintains roughly the same parameter count of standard embedding-based methods. The proposed approach can be also incorporated into other architectures by, e.g., changing the grounding of the predicates. Experimental results prove that FLVN achieves performance on par or exceeding that of current literature on common GZSL benchmarks.

We believe that our approach can be further extended in several ways. For example, different formulations of the `isOfClass` predicate, or the introduction of a `hasAttribute` predicate to predict image-level attributes, could incorporate attention to discriminative regions within the image and increase explainability. Introducing attention mechanisms based on object detection [4] could support the definition useful axioms that identify the most discriminative parts of the image. Introducing axioms that solely consider the similarity between images (with no knowledge of their class membership) could improve results for unseen classes without requiring additional labels in the dataset. The knowledge base could be also extended to consider other types of relationships, beyond class

hierarchies, and from emerging sources, such as language models. Finally, the proposed method could be explored in a trasductive ZSL setting.

Acknowledgements

This study was carried out within the FAIR - Future Artificial Intelligence Research and received funding from the European Union Next-GenerationEU (PIANO NAZIONALE DI RIPRESA E RESILIENZA (PNRR) – MISSIONE 4 COMPONENTE 2, INVESTIMENTO 1.3 – D.D. 1555 11/10/2022, PE00000013). This manuscript reflects only the authors' views and opinions, neither the European Union nor the European Commission can be considered responsible for them.

References

1. Yu, D., Yang, B., Liu, D., Wang, H., Pan, S.: Recent advances in neural-symbolic systems: A survey. arXiv preprint arXiv:2111.08164 (2021)
2. Badreddine, S., Garcez, A.d., Serafini, L., Spranger, M.: Logic tensor networks. Artificial Intelligence **303**, 103649 (2022)
3. van Bekkum, M., de Boer, M., van Harmelen, F., Meyer-Vitali, A., Teije, A.t.: Modular design patterns for hybrid learning and reasoning systems: a taxonomy, patterns and use cases. Applied Intelligence **51**(9), 6528–6546 (2021)
4. Manigrasso, F., Miro, F.D., Morra, L., Lamberti, F.: Faster-LTN: a neuro-symbolic, end-to-end object detection architecture. In: International Conference on Artificial Neural Networks. Springer (2021)
5. Donadello, I., Serafini, L.: Compensating supervision incompleteness with prior knowledge in semantic image interpretation. In: 2019 International Joint Conference on Neural Networks (IJCNN) (2019)
6. Xian, Y., Lampert, C.H., Schiele, B., Akata, Z.: Zero-shot learning — A comprehensive evaluation of the good, the bad and the ugly. IEEE Transactions on Pattern Analysis and Machine Intelligence (9) (2019)
7. Martone, S., Manigrasso, F., Lamberti, F., Morra, L.: PROTOtypical logic tensor networks (PROTO-LTN) for zero shot learning. 2022 26th International Conference on Pattern Recognition (ICPR) (2022)
8. Carraro, T.: LTNtorch: PyTorch implementation of Logic Tensor Networks (mar 2022), `https://doi.org/10.5281/zenodo.6394282`
9. Li, Z., Stengel-Eskin, E., Zhang, Y., Xie, C., Tran, Q.H., Van Durme, B., Yuille, A.: Calibrating concepts and operations: Towards symbolic reasoning on real images. In: IEEE/CVF International Conference on Computer Vision (2021)
10. Xu, W., Xian, Y., Wang, J., Schiele, B., Akata, Z.: Attribute prototype network for zero-shot learning. ArXiv (2020)
11. Cheng, D., Wang, G., Wang, N., Zhang, D., Zhang, Q., Gao, X.: Discriminative and robust attribute alignment for zero-shot learning. IEEE Transactions on Circuits and Systems for Video Technology (2023)
12. Zhang, L., Xiang, T., Gong, S.: Learning a deep embedding model for zero-shot learning. In: IEEE Conference on Computer Vision and Pattern Recognition (2017)

13. Zhu, P., Wang, H., Saligrama, V.: Generalized zero-shot recognition based on visually semantic embedding. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (2019)
14. Zhang, L., Xiang, T., Gong, S.: Learning a deep embedding model for zero-shot learning. IEEE/CVF Conference on Computer Vision and Pattern Recognition (2017)
15. Chen, L., Zhang, H., Xiao, J., Liu, W., Chang, S.F.: Zero-shot visual recognition using semantics-preserving adversarial embedding networks. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (2018)
16. Zhang, L., Wang, P., Liu, L., Shen, C., Wei, W., Zhang, Y., van den Hengel, A.: Towards effective deep embedding for zero-shot learning. IEEE Transactions on Circuits and Systems for Video Technology (2018)
17. Jiang, H., Wang, R., Shan, S., Chen, X.: Transferable contrastive network for generalized zero-shot learning. 2019 IEEE/CVF International Conference on Computer Vision (ICCV) (2019)
18. Li, Y., Liu, Z., Yao, L., Wang, X., Wang, C.: Attribute-modulated generative meta learning for zero-shot classification. ArXiv (2021)
19. Xie, G., Liu, L., Jin, X., Zhu, F., Zhang, Z., Qin, J., Yao, Y., Shao, L.: Attentive region embedding network for zero-shot learning. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2019)
20. Yu, Y., Ji, Z., Han, J., Zhang, Z.: Episode-based prototype generating network for zero-shot learning. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2019)
21. Liu, Z., Li, Y., Yao, L., Wang, X., Long, G.: Task aligned generative meta-learning for zero-shot learning. In: AAAI Conference on Artificial Intelligence (2021)
22. Li, J., Jing, M., Lu, K., Ding, Z., Zhu, L., Huang, Z.: Leveraging the invariant side of generative zero-shot learning. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2019)
23. Felix, R., Reid, I., Carneiro, G., et al.: Multi-modal cycle-consistent generalized zero-shot learning. In: Proceedings of the European conference on computer vision (ECCV). pp. 21–37 (2018)
24. Han, Z., Fu, Z., Chen, S., Yang, J.: Contrastive embedding for generalized zero-shot learning. 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2021)
25. Van der Maaten, L., Hinton, G.: Visualizing data using t-sne. Journal of Machine Learning Research (11) (2008)
26. Chen, S., Hong, Z., Xie, G., Wang, W., Peng, Q., Wang, K., et al.: MSDN: Mutually semantic distillation network for zero-shot learning. 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2022)
27. Wang, W., Li, Q.: Generalized zero-shot activity recognition with embedding-based method. ACM Transactions on Sensor Networks (2023)
28. Wah, C., Branson, S., Perona, P., et al.: Multiclass recognition and part localization with humans in the loop. International Conference on Computer Vision (2011)
29. Xiao, J., Hays, J., Ehinger, K.A., Oliva, A., Torralba, A.: SUN database: Large-scale scene recognition from abbey to zoo. In: IEEE Conference on Computer Vision and Pattern Recognition (2010)
30. Sikka, K., Huang, J., Silberfarb, A., Nayak, P., Rohrer, L., Sahu, P., et al.: Zero-shot learning with knowledge enhanced visual semantic embeddings. ArXiv (2020)
31. Su, H., Li, J., Lu, K., Zhu, L., Shen, H.T.: Dual-aligned feature confusion alleviation for generalized zero-shot learning. IEEE Transactions on Circuits and Systems for Video Technology (2023)