Semantic Web 1 (0) 1–5
IOS Press

Improving Image Classification for Geospatial Data using a Semantic Referee

Alirezaie Marjan [a,*], Längkvist Martin [a] Sioutis Michael [a] and Loutfi Amy [a]
[a] *Center for Applied Autonomous Sensor Systems, Örebro University, Örebro, Sweden*

Editors: First Editor, University or Company name, Country; Second Editor, University or Company name, Country
Solicited reviews: First Solicited Reviewer, University or Company name, Country; Second Solicited Reviewer, University or Company name, Country
Open reviews: First Open Reviewer, University or Company name, Country; Second Open Reviewer, University or Company name, Country

Abstract.
Recent machine learning algorithms have shown a considerable success in various computer vision tasks, including semantic segmentation. However, they seldom perform without error. A key aspect of discovering why the algorithm has failed is usually the task of the human who, using domain knowledge and contextual information, can discover systematic shortcomings in either the data or the algorithm. In this paper, we propose a symbolic-based technique, called a semantic referee, which is both able to explain the errors emerging from the machine learning framework and suggest corrections. The semantic referee relies on a spatial reasoning method applied on ontological knowledge in order to retrieve the features of the errors in terms of their spatial relations with their environment. The symbolic explanation of the errors is then reported to the learning algorithm to learn from its mistakes and consequently improve the performance. In this paper, the proposed method of the interaction between a neural network classifier and a semantic referee show how to improve the performance of semantic segmentation for satellite imagery data.

Keywords: Deep Neural Network, Semantic Referee, Spatial Reasoning, Geo-Ontology, OntoCity

1. Introduction

Machine learning algorithms and semantic web technologies have both been widely used in geographic information systems. The former is typically applied on geo-data to perform image recognition tasks including object recognition. While the latter is used for a number of applications ranging from localization, navigation knowledge acquisition and map query [1]. Despite recent success in machine learning, in particular, with deep learning methods for image segmentation and classification of satellite data, seldom do ML approaches take into account the advantages of semantics associated to geo-data. Rather, machine learning algorithms are typically trained by optimizing a cost function that continuously measures the training errors during learning and adapt the model parameters in order to minimize these errors. Seemingly, these algorithms learn from their errors; however, such learning processes differ from what human advisors usually mean by "*learn-from-your-mistakes*", which entails that the learner is able to understand why the errors occurred and conceptualize them by expressing their characteristics. The training process of minimizing a cost function is not aimed towards explaining the errors themselves or describing why such errors have been made in the first place, but instead follows some rules for parameter updates that are predefined by the selected optimization method.

In the context of semantic segmentation and classification for geospatial data, a classifier that uses only the RGB channels as input runs the risk of producing a large amount of misclassifications (errors) due to the visual similarity between certain classes. For

*Corresponding author. E-mail: firstname.lastname@oru.se.

example, in satellite imagery, the RGB channels *water* looks similar to *shadows*, and *buildings* with gray roofs look similar to *roads*. One solution to this problem, which has been addressed in previous works, is to use additional sources of information as input to the classifier, such as Synthetic-aperture radar (SAR), Light detection and ranging, (LIDAR), or Digital Elevation Model (DSM) for the height information, and/or hyperspectral bands, near-infrared (NIR) bands, and synthetic spectral bands for texture and color information [2, 3]. However, such data is not always accessible, e.g., satellite images that only contain RGB channels from Google Maps and similar sources. Another possible solution to increase the performance of the classifier is to change the architecture to increase the capacity, e.g., by using Deep Convolutional Neural Networks (DCNNs) [4–6].

In this paper, instead of relying on additional sources of information, which can be hard to acquire and/or integrate, or taking the ad-hoc approach of experimenting with the architecture of the classifier, we focus on explaining the errors in terms of their spatial relations and neighborhood instead, as a means to ameliorate the performance of machine learning algorithms. To this end, we propose a representation of the context that includes symbolic concepts and their relations, in order to reason upon and retrieve the required characteristics of the data. We refer to this process as a semantic referee as knowledge representation and reasoning methods are used to arbitrate on the errors arising from misclassifications (errors).

In particular, our representation makes use of RCC-8 spatial relations, as well as extensions thereof, where RCC-8 stands for the language that is formed by the 8 base relations of the Region Connection Calculus [7], viz., *disconnected*, *externally connected*, *overlaps*, *equal*, *tangential proper part*, *non-tangential proper part*, *tangential proper part inverse*, and *non-tangential proper part inverse*. Notably, RCC-8 has been adopted by the GeoSPARQL[1] standard, and has found its way into various Semantic Web tools and applications over the past few years [8]. A worth-mentioning cross-disciplinary application of RCC-8 reasoning involves segmentation error correction for images of hematoxylin and eosin (H&E)-stained human carcinoma cell line cultures [9]. In our work, inspired by the integration of qualitative spatial reasoning methods into imaging procedures as described in [9] (or into any other domain for that matter), we aim to employ qualitative spatial reasoning techniques in order to assist deep learning methods for image classification via interaction and guidance.

Reconciliation of data-driven learning methods with symbolic reasoning has been identified in the literature as one of the key challenges in Artificial Intelligence [10]. Depending on the approaches to represent both low and high level data, such approaches have been addressed under different names that include abduction-induction in learning [11], structural alignment [12], and neural-symbolic methods [13, 14]. With the increasing interest in connectionist learning systems, and particular in deep learning methods, research on designing neural-symbolic systems has recently made considerable progress. Such systems are routinely referred to as Explainable Artificial Intelligence (XAI), and used to provide better insights into the learning process [15].

1.1. Contribution

In this work, we propose an ontology-based reasoning approach to assist a neural network classifier for a semantic segmentation task. This assistance can be used in particular to represent typical errors, provide possible explanations, and eventually assist in correcting misclassification. We show using a specific case of on large scale satellite data how semantic web resources interacts with deep learning to improve the classification performance and explainability of a system on a city wide scale.

Our contribution differentiates from the neural-symbol systems explained in Section 2 in three regards. Firstly, our method plays the role of a semantic referee for the imagery data classifier in order to explain its errors, which, to the best of our knowledge, is the first attempt in the domain of image segmentation to tackle the problem by explaining it. Secondly, our model focuses on the misclassifications and uses ontological knowledge, with concepts and their spatial relations, together with a geometrical processing to explain them. Finally, our system closes the communication loop between the classifier and the semantic referee, which enables the classifier to learn how to prevent making the same mistakes.

1.2. Structure of paper

The rest of the paper is structured as follows. Section 2 describes the related work. The method is pre-

[1] http://www.opengeospatial.org/standards/geosparql

sented in Section 3, which gives the overview of the approach (Section 3.1), the satellite image data used in this work (Section 3.2), the neural network-based semantic segmentation algorithm (Section 3.3), the OntoCity as the ontological knowledge model (Section 3.4), and the error explanation process and how this explanation is used to guide the classifier (Sections 3.5 and 3.6). The experimental evaluation is presented in Section 4, which is followed by a discussion and possible directions for future work in Section 5.

2. Related Work

As discussed in [16], in neural-symbolic systems where the learning is based on a connectionist learning system, one way of interpreting the learning process is to explain the classification outputs using the concepts related to the classifier's decision. However, there is a limited body of work where symbolic techniques are used to explain the conclusions. The work presented in [17] introduces a learning system based on a Long-term Convolutional Network (LTCN) [18] that provides explanations over the decisions of the classifier. An explanation is in the form of a justification text. In order to generate the text, the authors have proposed a loss function upon sampled concepts that, by enforcing global sentence constraints, helps the system to construct sentences based on discriminating features of the objects found in the scene. However, no specific symbolic representation was provided, and the features related to the objects are taken from the sentences that are already available for each image in the dataset (CUB dataset [19]).

With focus on the knowledge model, the work presented in [20] proposes a system that explains the classifier's outputs based on the background knowledge. The key tool of the system, called DL-Learner, works in parallel with the classifier and accepts the same data as input. Using the Suggested Upper Merged Ontology (SUMO)[2] as the symbolic knowledge model, the DL-Learner is also able to categorize the images by reasoning upon the objects together with the concepts defined in the ontology. The compatibility between the output of the DL-Learner and the classifier can be seen as a reliability support and at the same time as an interpretation of the classification process.

Similarly, the work detailed in [21] relies on a general-purpose knowledge model called the ConceptNet Ontology, where the integration of the symbolic model and a sentence-based image retrieval process based on deep learning is used to improve the performance of the learning process. The knowledge about different concepts, e.g., their affordances, their relations with other objects, is aligned with objects derived from the deep learning method.

Although in the these works the role of the symbolic knowledge represented by ontologies in regard to improving or interpreting the learning process has been emphasized, they are limited in terms of the symbolic representation models. More specifically, the concepts and their relations in ontologies are simplified, limiting the richness of deliberation in an eventual reasoning process, especially for visual imagery data.

3. Method

3.1. Overview of the approach

An overview of our approach can be seen in Figure. 1, which shows the interaction between the semantic referee and the classifier. In order to deal with the misclassifications (errors) made by an imagery data classification method, we use a semantic referee that is able to make sense of the errors and consequently provide more useful information for the classifier to learn from its mistakes.

The process of making sense of the errors includes the conceptualization of the misclassified areas based on their physical (e.g., geometrical) properties. The conceptualization process is performed by a spatial reasoner associated with the ontological knowledge. The reasoner first extracts the geometrical properties of the given misclassified area (e.g., *a building and a road are connected to the misclassified area*) and then aligns these features with the available ontology. The reasoner eventually infers the best possible match for the error w.r.t the available ontological knowledge. The inferred concept related to the misclassified area is then given to the classifier as a referee providing information to be used at the next rounds of learning.

3.2. Data

The data used in this work consists of a RGB satellite image of central Stockholm, Sweden, shown in Figure. 2. The selected area size is 4000×8000 pixels with a pixel-resolution of 0.5 meters and was divided into train and test sets with a $50 - 50$ split. The ground

[2] http://www.adampease.org/OP/

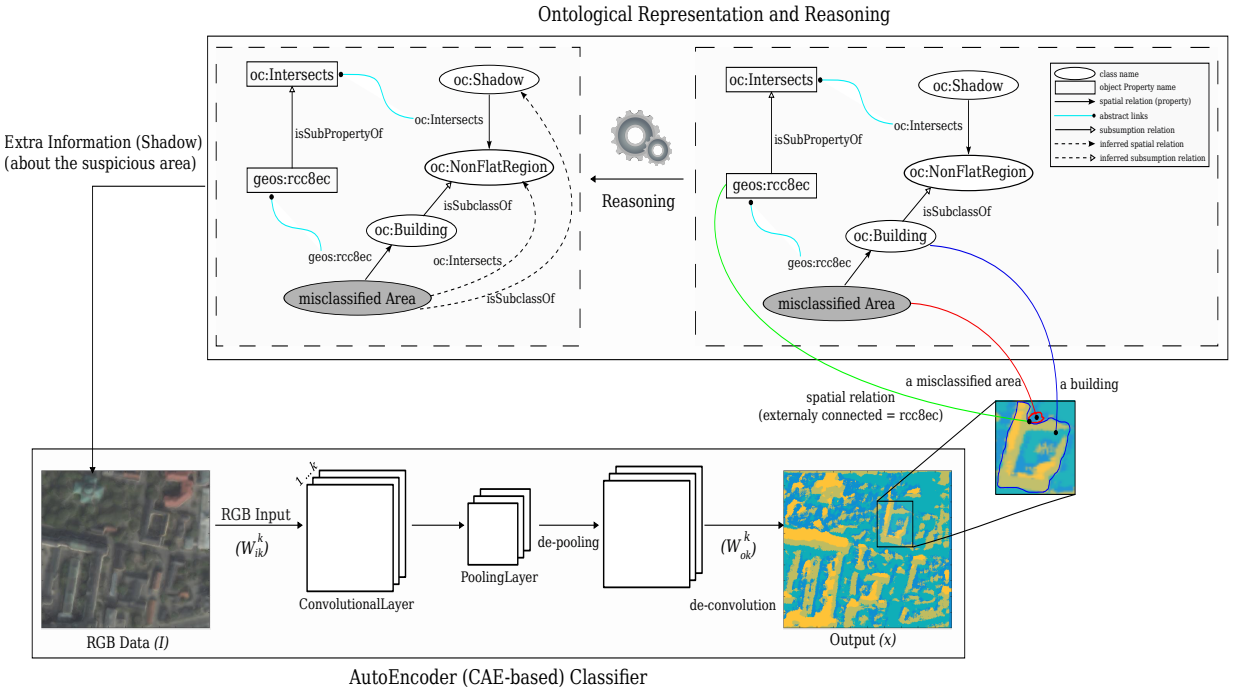

Fig. 1. Overview of applying a semantic referee (top layer) in the form of spatial reasoning upon ontological knowledge to improve the image segmentation task of the classifier (bottom layer). The classifier consists of a Convolutional Auto-Encoder (CAE), which receives the RGB data to perform the semantic segmentation. The semantic referee explains the mistakes made by the classifier based on ontological concepts and feed it back to the classifier to avoid it making the same mistakes. The ontological knowledge and reasoning methods that play the role of semantic referee to make sense of errors and explain them. For example, the *misclassified area* (in red) is *externally connected* (`geos:rcc8ec`) to the *building* region (in blue). By mapping the 3 entities into their equivalent concepts in the ontology, the ontological reasoner infers the direct superclass of the misclassified area, which is the class `oc:shadow` whose constraints are more general ($\exists$ `oc:intersects.oc:NonFlatRegion`) than the spatial representation of the red misclassified area.

truth used for supervised training and evaluation has been provided by Lantmäteriet, the Swedish Mapping, Cadastral and Land Registration Authority. The 5 categories that are used (and their prevalence in percentages) are *vegetation* (7.6), *road* (31.3), *building* (35.4), *water* (23.7), and *railroad* (2.2). Due to the large imbalance in the data set, the training data was oversampled and the test set was undersampled to contain an equal amount of pixels for each of the 5 classes.

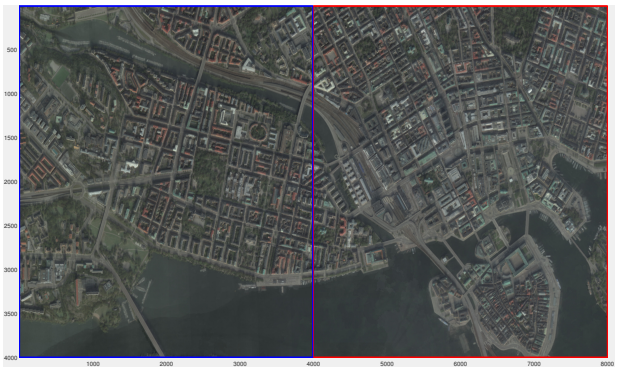

Fig. 2. RBG satellite image of Stockholm, Sweden, that was used for training (right side) and testing (left side).

3.3. Data classification

A Convolutional Auto-encoder (CAE) [22] is used to perform the semantic segmentation where every pixel in the map is classified. Each layer in the CAE consists of an encoder, that performs convolution and pooling, and a decoder, that performs unpooling and deconvolution.

The convolutional layer in the encoder calculates the k-th feature map as:

$$h^k = \sigma\left(I^i * W^k_{ik} + b^k\right) \quad (1)$$

where I^i is the input image with color channel i, W_{ik}^k is the k-th filter from input channel i and filter k, b^k is the bias for the k-th filter, σ is the Rectified Linear Unit (ReLu) [23] non-linear activation function, and $*$ denotes the convolution operation. The pooling layer is calculated by downsampling the convolutional layer by taking the maximum value in each $p \times p$ non-overlapping subregion.

In the decoder, an unpooling process and deconvolution is performed. The unpooling is performed with switch variables [24] that remember the position of the maximum value during the pooling operation. The deconvolution is performed to obtain the final output, x, which is calculated as:

$$x = softmax\left(o^k * W_{ok}^k + c^K\right) \quad (2)$$

where o^k is the k-th map of the unpooling layer, W_{ok}^k is the k-th filter from unpooling layer o and filter k, and c^K is the bias for the K-th output layer.

The architecture for the CAE used in this work consist of a 5-layer CAE with filter sizes $[11, 9, 7, 5, 3]$ and the number of filters is $[10, 20, 30, 40, 50]$ for each layer, respectively. Max-pooling with pooling dimension 2 is used in each layer. The ReLu-activation function is used in each layer except for last layer that uses a softmax activation function. The parameters were initialized with Xavier initialization [25] and trained using the AdaGrad optimization method [26].

3.4. OntoCity: the ontological knowledge model

In our approach the improvement of data classification relies on a spatial reasoning process applied upon ontological knowledge. The ontology that we have used as the knowledge model is called OntoCity[3] which contains the domain knowledge about generic spatial constraints in outdoor environments. OntoCity whose (part of) representational details can also be found in [27] is an extension of the GeoSPARQL[4] ontology which is known as a standard vocabulary for geospatial data. The main idea behind designing OntoCity was developing a generalized knowledge model to represent cities in terms of their structural, conceptual and physical aspects as well as their types (e.g., natural or man-made) and their relations (e.g., spatial constraints, affordances, etc.). Figure 3 illustrates a Protégé [28] snapshot of the hierarchy of concepts defined in OntoCity.

[3]https://w3id.org/ontocity/ontocity.owl
[4]http://www.opengeospatial.org/standards/geosparql

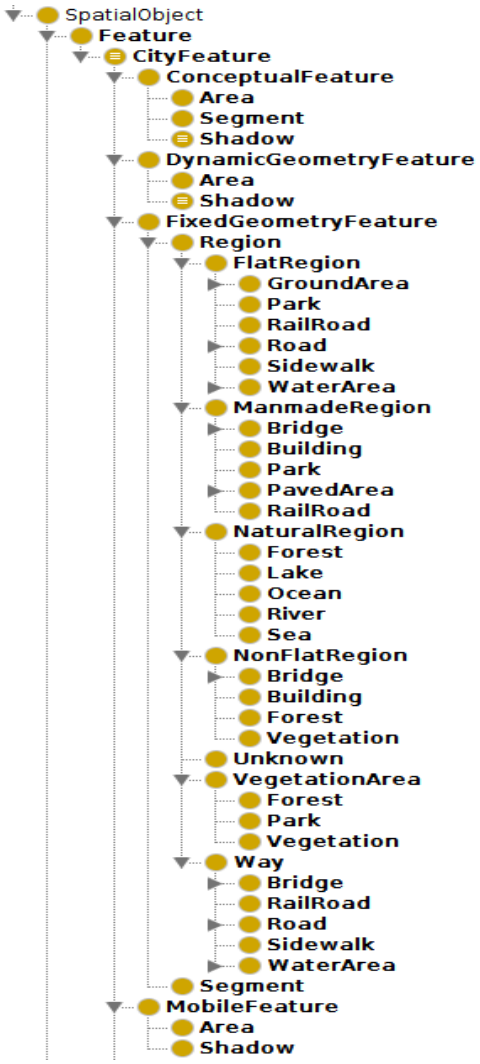

Fig. 3. The snapshot of the city features' subsumption relations in OntoCity. The city features are defined as the subclasses of the `geos:Feature` class defined in GeoSPARQL.

The class `oc:CityFeature` is one of the general classes defined in OntoCity, and is subsumed by the concept `geos:Feature` in GeoSPARQL. As you can see, the name of a class has a prefix which indicates the ontology that it belong to. In the aforementioned classes, the two prefixes `oc` and `geos` refer to the URIs of OntoCity and GeoSPARQL, respectively.

The class `geos:Feature` that represents any spatial object with a geometry is specialized into the class `oc:CityFeature` which represents such features in a city that are in the form of polygons and are at least with one spatial relation with the other city features.

The axioms of OntoCity given in this paper are in description logic (DL) [29]:

$$\texttt{oc:CityFeature} \sqsubseteq \texttt{geos:Feature} \sqcap \exists\, \texttt{geos:hasGeomtery.geos:Polygon} \sqcap \exists\, \texttt{oc:hasSpatialRelation.oc:CityFeature}$$

Spatial relations in OntoCity includes the RCC-8 (Region Connection Calculus) [7] relations defined in [7] and adopted by GeoSPARQL, with a bit of extension. The extension includes the definition of the relation `oc:intersects` that subsumes several RCC-8 relations including partially overlapping (`geos:rcc8po`) and externally connected (`geos:rcc8ec`). The spacial relation `oc:intersects` is used to simplify the representation of some situations for which we only need to know whether the two city features are intersecting and no matter how.

Spatial relations are used in the form of spatial constraints to provide meanings to the city features. City features are categorized into several types that are defined as the direct subclasses of `oc:CityFeature` in OntoCity. These categories include `oc:PhysicalFeature` and `oc:ConceptualFeature`, that represents features with physical geometry (e.g., a landmark with an absolute elevation value measured from the sea floor), or conceptual geometry (e.g., a rectangular division in a city regardless of their landmarks), respectively. Furthermore, the two other classes `oc:FixedGeometryFeature` and `oc:DynamicGeometryFeature` exist to represent features whose geometries are fixed or dynamic (changing in time). Mobility is another property that can categorize the city features into mobile (`oc:MobileFeature`, e.g., a car), or stationary (`oc:StationaryFeature`, e.g., a building). The following axioms show the subsumption relations between `oc:CityFeature` and the aforementioned classes:

$$\texttt{oc:DynamicGeometryFeature} \sqsubseteq \texttt{oc:CityFeature}$$
$$\texttt{oc:FixedGeometryFeature} \sqsubseteq \texttt{oc:CityFeature}$$
$$\texttt{oc:MobileFeature} \sqsubseteq \texttt{oc:CityFeature}$$
$$\texttt{oc:StationaryFeature} \sqsubseteq \texttt{oc:CityFeature}$$
$$\texttt{oc:ConceptualFeature} \sqsubseteq \texttt{oc:CityFeature}$$
$$\texttt{oc:PhysicalFeature} \sqsubseteq \texttt{oc:CityFeature} \sqcap \exists\, \texttt{oc:hasAbsoluteElevationValue.xsd:double}$$

As shown in Figure. 3, each of the subclasses of the class `oc:CityFeature` has its own taxonomy. For example, the class `oc:Region` as a *physical* feature with a *fixed geometry* which is also *stationary* (i.e., non-mobile) represents a landmark that can per se be categorized into various types such as flat or non-flat, or likewise, into man-made or natural regions:

$$\texttt{oc:Region} \sqsubseteq \texttt{oc:PhysicalFeature} \sqcap \texttt{oc:StationaryFeature} \sqcap \texttt{oc:FixedGeometryFeature}$$
$$\texttt{oc:ManmadeRegion} \sqsubseteq \texttt{oc:Region}$$
$$\texttt{oc:NaturalRegion} \sqsubseteq \texttt{oc:Region}$$
$$\texttt{oc:FlatRegion} \sqsubseteq \texttt{oc:Region}$$
$$\texttt{oc:NonFlatRegion} \sqsubseteq \texttt{oc:Region} \sqcap \exists\, \texttt{oc:hasRelativeElevationValue.xsd:double} \sqcap \exists\, \texttt{oc:intersects.oc:Shadow}$$

For each location in a city (or in general on the ground) there are two elevation value, namely absolute elevation and relative elevation. The absolute elevation is the value measured from the sea-level with height value zero, whereas the relative elevation value of a specific location indicates its relative hight w.r.t the ground level and its vicinity. By a non-flat region we refer to landmarks of a city with a non-zero relative elevation value. Due to its height, a non-flat region is also assumed to cast shadows. As we will see, the concept shadow has been also defined in OntoCity (`oc:Shadow`) due to its spatial relations with the other city features.

The texture of regions (i.e., landmarks) are defined as subclasses of the class `oc:Region`. It is worth mentioning that some of these region types are equivalent to the labels taken into account by the classifier to classify regions. These regions are defined as follows:

$$\texttt{oc:River} \sqsubseteq \texttt{oc:WaterArea} \sqsubseteq \texttt{oc:Region}$$
$$\texttt{oc:Road} \sqsubseteq \texttt{oc:PavedArea} \sqsubseteq \texttt{oc:ManmadeRegion}$$
$$\texttt{oc:Park} \sqsubseteq \texttt{oc:VegetationArea} \sqsubseteq \texttt{oc:Region}$$
$$\texttt{oc:Building} \sqsubseteq \texttt{oc:ManmadeRegion} \sqcap \texttt{oc:NonFlatRegion}$$

The RCC-8 relations are used to describe more specific features (e.g., bridges, shadows, shores) whose definitions rely on their spatial relations with their vicinity. For instance, a bridge is a man-made, non-flat region which is partially overlapping (referring to the RCC-8 relation `geos:rcc8po`) at least one region

whose texture identify the bridge type. If the region is a water-area then the overlapping bridge is a water bridge, or if the region is a street, then the bridge is categorized as a street or a pedestrian bridge:

$$\texttt{oc:Bridge} \sqsubseteq \texttt{oc:ManmadeRegion} \sqcap \texttt{oc:NonFlatRegion} \sqcap \exists\, \texttt{geos:rcc8po.oc:Region}$$

The concept shadow as a spatial feature with a geometry is also defined in OntoCity. Although the shape of shadows depends on the exact position of the source light and also the height value of the casting objects, it is still possible to qualitatively describe shadows in the ontology. In OntoCity, a shadow is seen as a conceptual (non-physical) feature whose geometry is dynamic and mobile (i.e., changing depending on the time of the day). The definition of the concept shadow becomes more precise by adding the spatial constrains saying that a shadow is also intersecting (`oc:intersects`) with at least one non-flat region (likely as its casting object):

$$\texttt{oc:Shadow} \sqsubseteq \texttt{oc:ConceptualFeature} \sqcap \texttt{oc:DynamicGeometryFeature} \sqcap \texttt{oc:MobileFeature} \sqcap \exists\, \texttt{oc:intersects.oc:NonFlatRegion}$$

The aforementioned axioms in OntoCity were a subset of general knowledge (e.g., "*Water bridges cross water areas*") that always hold in the context of any city. However, the background knowledge can be much more specific and identify features of a specific environment (e.g., "*in the given region there is no building connected to water areas*").

3.4.1. Specialization of OntoCity

The area under our study, as shown in Section 3.2, belongs to the central part of Stockholm. Knowing this area, we could specialize the definition of some of its regions by adding more spatial constraints as follows:

1. Buildings are directly connected to at least a road or a vegetation area (referring to the connected relation in RCC8: `geos:rcc8ec` relation)
2. Buildings are not intersecting with railroads (referring to the negation of the `oc:intersects` relation)
3. Buildings are not directly connected to water-area (referring to the negation of externally connected relation in RCC8: `geos:rcc8ec`).
4. Buildings are not contained by roads (referring to the negation of tangential proper part relation in RCC-8: `geos:rcc8tpp`).
5. Buildings do not contain roads (referring to the negation of tangential proper part inverse relation in RCC-8: `geos:rcc8tppi`).
6. Railroads are not directly connected to water-area (referring to the negation of the `oc:intersects` relation).

The following axiom shows the DL definition of the class `oc:StockholmBuilding` as the subclass of the class `oc:Building`:

$$\texttt{oc:StockholmBuilding} \sqsubseteq \texttt{oc:Building} \sqcap \exists\, \texttt{geos:rcc8ec.(oc:VegetationArea} \sqcup \texttt{oc:Road)} \sqcap \nexists\, \texttt{oc:intersects.oc:RailRoad} \sqcap \nexists\, \texttt{geos:rcc8ec.oc:Waterarea} \sqcap \nexists\, \texttt{geos:rcc8tpp.oc:Road} \sqcap \nexists\, \texttt{geos:rcc8tppi.oc:Road}$$

The spatial constraints used in the definition of classes are considered by a reasoner in order to discard the impossible labels (region types) for a region based on its neighborhood.

3.5. Explaining misclassifications

Given the classification outputs which include both the classified and misclassified regions together with ontological knowledge about city features, the (spatial/ontological) reasoner as a semantic referee is able to explain the errors based on the content of the ontology. The process is composed of several steps which are in brief captured in Algorithm 1.

Algorithm 1 Error Explanation

Require: $S = empty, m, R$
▷ S: A hash-map, empty in the beginning
▷ P: The given list of misclassified areas
▷ R: The given list of classified regions
for each $r \in \mathcal{R}$ **do**
  $t \leftarrow getRegionType(r)$
  for each $p \in \mathcal{P}$ **do**
    $q \leftarrow calculateRCC(p, r)$
    $S.add(< q, t >)$
  end for
end for
$< Q, T > \leftarrow getHighFrequentSpatialRelation(S)$
$C \leftarrow queryOntology(Q, T)$
$explanation \leftarrow getSemantics(C)$

The output of the classifier is in the form of labeled pixels where each pixel is also assigned with a classification certainty probability. The pixels with low certainty are suspected to be misclassified ones and should be prioritized for inspection by the reasoner.

The algorithm accepts as input the list of segments (polygons) for both classified ($\mathcal{R}$) and misclassified areas ($\mathcal{P}$). Given the two polygon lists of $\mathcal{R}$ and $\mathcal{P}$, the algorithm calculates all the possible (RCC-8) qualitative spatial relations between any pairs of (p, r) where $p \in \mathcal{P}$ is a misclassified area and $r \in \mathcal{R}$ is a classified region in its vicinity. For each pair (p, r), besides the calculated spatial relation q, the algorithm also keeps the type of the region r named as t. This information for each pair is added to the list S, which at the end of the geometrical calculation process will contain all the spatial relations that exist between the misclassified areas for each specific region type (see lines 4-10). At the end of the geometrical processing, S contains the geometrical characteristics of the misclassified areas.

As the next step, to find a general description indicating why the classifier has been confused, the characteristics of the errors are generalized based on their frequency. If we assume that the pair $< Q, T >$ (see line 11) represents the most observed spatial relation Q between the misclassified areas and a specific region type T, then this pair can be generalized and counted as a representative feature of the misclassified areas.

Given the representative pair $< Q, T >$, the algorithm continues to query OntoCity and asks for all the spatial features of the city that are at least in one Q relation with type T. The DL expression of the query is: $\exists\, T.Q$.

By applying the ontological reasoner the query can also be further generalized from type T to its superclasses in OntoCity (see line 12). The concept (C) as a spatial feature ($C \sqsubseteq$ `oc:CityFeature`) inferred by the reasoner, is considered as the explanation pertaining the errors.

3.6. Improving the classifier with the explanations

One of the major challenges for neural-symbolic approaches is to make the two systems communicate in a way that they understand each other. There are a number of ways that the output from the reasoner (i.e., the error explanation) can influence a neural network-based classifier, e.g., training set selection, data selection, architecture design, and cost function modification. In this work, our strategy to establish an interaction between the classifier and the reasoner is to provide the classifier with additional information that is generated by the reasoner. This information is represented as additional input channels to the classifier beside the standard RGB channels.

Introducing new channels of data to the classifier is not as easy as relabeling the misclassified areas with the explanation that the semantic referee has found. Although the explanation is based on the features of the misclassified areas, it only helps us to make sense of the errors, or find the main cause behind the misclassification.

The interaction of the reasoner with the classifier (i.e., providing channels of data) highly depends on the semantics of the inferred explanation. As we will see in Section4, the reasoner finds shadows as the main cause behind the misclassification of our satellite imagery data. In order to report the concept of shadow back to the classifier, we first need to localize them on the map. Although neither in OntoCity nor in other available ontologies are there any formal representation to calculate the location of shadows, as we will see in Section 4.2, this explanation as a semantic referee provides a significant insight for us to develop the geometrical reasoner to localize the shadows.

4. Empirical Evaluation

4.1. Reasoner explaining errors

The first step for reasoning about the errors is to locate the regions that are misclassified. Since the ground truth is available for our data, the misclassified areas are easy to locate. The second step is to select which regions should be sent to the reasoner. In this work, we use the classification certainty and select the top 20 regions for each of the 5 classes (totally 100 regions) that the classifier is mostly unsure about. Given the classified regions and the misclassified areas, as explained in Section 3.5, the spatial reasoner together with the ontological knowledge are responsible for explaining the errors.

The explanation process is based on the spatial features of the errors. Therefore, the spatial reasoner first takes into account the top 100 suspected misclassified areas to extract their spatial relations with their segmented neighborhood. This step has been implemented using the open-source JTS Topology Suite[5], whose summary of results are shown in Table 1. Each

[5]https://github.com/locationtech/jts

cell of the table represents number of misclassified areas that are in a special relation (given in the column header) with all the regions with a specific type (given in the row header).

To find a representative feature of the misclassified areas, Algorithm 1 considers the pair $< Q, T >$ as the most observed spatial relation Q between the misclassified areas and a specific region type T, which in our case, as shown in Table 1, is the pair $< Q=$ `geos:rcc8ec`, $T=$ `oc:Building`$>$ which involves 136 misclassified areas.

Type (t) \ Relation (q)	ec	po
oc:Building	**136**	3
oc:Road	59	0
oc:Water	11	0

Table 1

Summary of the the spatial features of misclassifications categorized based on their region types: Each cell value represents the number of misclassified regions involved in the given spatial relations (q) with the given region type (t), where `ec` and `po` refer to the RCC-8 relation *externally connected* and *partially overlapping*, respectively.

The pair $< Q, T >$ is enough to query the ontology with spatial constraints. We have extended and used the reasoner Pellet, as an open-source Java based OWL 2 ontological reasoner [30]. The extension is in terms of filtering concepts based on their spatial constraints.

The Description Logic (DL) syntax of the query given to the reasoner is $\exists$ `geos:rcc8ec.oc:Building` interpreted as "*all the entities that are at least in one* `geos:rcc8ec` *relation with the region type* `oc:Building`". The ontological reasoner results in a hierarchically linked concepts in the ontology from the most generalized to the most specialized (direct superclass) concepts satisfying the constraint given in the query. The satisfactory concept is explained as "*a mobile conceptual feature with a dynamic geometry*" or more specifically a `oc:shadow` (as a direct answer of the query). In OntoCity, the concept shadow is defined based on the spatial constraint: $\exists$ `oc:intersects.oc:NonFlatRegion`, which is found by the reasoner as the generalization of the query $\exists$ `geos:rcc8ec.oc:Building` (where, `geos:rcc8ec` $\sqsubseteq$ `oc:intersects` and `oc:Building` $\sqsubseteq$ `oc:NonFlatRegion`) (see Figure. 1, top layer).

Figure 4 shows two samples taken from the classification output, with some marked misclassified areas. At the first row, the areas marked with number 1 and 2 are misclassified as water. As the RGB image on the left illustrates, the marked areas are connected to buildings which cast shadows. Knowing that an area is under shadow, we can explain that the classifier is confused due to the similarity between the color of the shadow and the color of water (both looked dark). At the second row, the area marked with number 1 is likewise misclassified as water. This area is again (externally) connected with a building whose shadow can explain the misclassification. This area is furthermore located between (i.e., connected with) at least two disconnected regions labeled as roads which are disconnected at the shadow area. It can explain the second most observed relation listed in Table 1, between the misclassified areas and the region type `oc:Road`.

Assuming buildings are often located close to roads (or streets), their shadow are likely casted on some parts of the roads. Therefore, a road instead of being recognized as a single road, is segmented into several roads disconnected at the shadow areas due to the change in their colors. Errors caused by shadows are not always labeled as water. Again in the second row, the areas marked with number 2 and 3 are also connected to buildings and roads, however, are misclassified as railroads again due to the fact that the darkness of the shadow at this location is similar to the captured color of railroads in the image data.

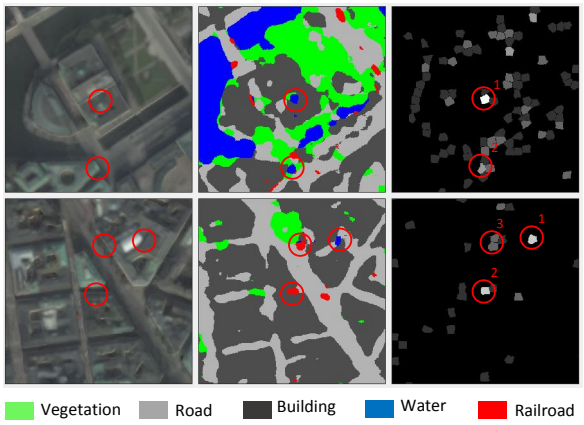

Fig. 4. Two examples of the classification output along with their input RGB image, classified segmentation and the misclassification. The misclassified areas marked with numbers are in spatial relations with buildings, roads, vegetation, etc. The ontological reasoner explains the misclassification as the result of the shadow of buildings on their neighborhood.

4.2. Shadow localization

Knowing that the main reason behind the misclassification, the semantic referee is expected to guide the classifier to better tackle its mistakes. For this, the location of shadows is an important factor that should be provided to the classifier. There is a fair amount of research work with the focus on shadow detection in the fields of computer vision and pattern recognition [31]. However, since the focus of this work is not to design an algorithm for shadow detection, but instead we are interested in informing the classifier of possible causes for the misclassification, it suffices to inform the classifier that there is a certain property about this region that differentiates it from other regions.

Another property that has an influence on the classification and might be a cause for the misclassifications is elevation. Since elevation difference of regions is one of the main parameters in casting shadows, we have assigned the relative elevation value for each region as the average of its pixels' elevation values. Given the elevation value together with the type and the spatial relations of regions in neighborhood of each misclassified area, the geometrical reasoner is able to localize the shadows as the group of pixels of the misclassified area with the lowest elevation value with respect to the elevation values of the regions intersecting with the misclassified area.

One of the strategy for representing the explanations from the reasoner to the classifier is to provide more channels of data. We have defined two channels where the first one represents the shadow pixels and the second describe an estimation of the relative height of the regions. These two channels of data are added to the RGB channels from the second round of learning.

4.3. Classification accuracy results

The classifier was trained on the training set and then applied on the test data. The classifier was first trained using only the RGB channels and empty (set to 0) channels for shadow and height estimation. On the subsequent 3 rounds, the classifier was then fine-tuned with the feedback from the reasoner in the form of adding information to the shadow and elevation channels. The overall classification accuracy on the test set was 75.8%, 79.1%, 82.8%, and 84.8% for each round, respectively.

The RGB inputs, predictions, shadow and height estimations can be seen in Figure. 5. The confusion matrix for the last round on the test set is also given in Table 2. As shown from the confusion matrix, the most difficult class to classify is the class *road* and the largest confusion is between *roads* and *buildings*. The second largest confusion is between *vegetation* and *roads*. Although the semantic referee has been able to improve the classification (from 75.8% to 84.8% accuracy), the confusions are still there. The reason behind the confusions regarding the class *vegetation* can be related to their wide range of elevation values. The label *vegetation* is not precise enough as it includes tress, lawns, parks, grass on the map and that is why their elevation values are not informative enough to help the classifier to differentiate them from roads.

		Predicted label				
	%	Vegetation	Road	Building	Water	Railroad
Actual label		87.5	6.4	4.9	0.4	0.7
		7.7	74.7	12.1	2.5	3.0
		4.3	9.6	84.7	1.2	0.1
		0.4	2.8	0.0	96.8	0.1
		0.6	4.4	1.0	0.1	93.8

Table 2

Confusion matrix [%] for the test set for the last round of the classifier before the reasoner has done any corrections.

The hardware that was used to train the classifier was a i7-8700K CPU @ 3.70Ghz with a GeForce GTX 1070 GPU. The initial training time for round 1 was around 72 hours and the training time for the subsequent rounds was around 24 hours for each round. Early-stopping was used on a validation set for deciding when to stop the training.

4.4. Reasoner correction results

We have so far shown how the semantic referee has found the reason behind the misclassification and reported it back to the classifier. However, one might ask why the reasoner instead of reporting the data back to classifier, does not correct the misclassifications using the spatial constraints given in the specialized version of OntoCity (see Sectionse 3.4.1). Although, the constrains in this ontology are not complete and are mainly about buildings, they are still useful to at least reduce the number of possible labels that the reasoner can assign to the areas under shadow. For instance, the labels of some of the misclassified areas shown in Figure 4 are inconsistent with the spatial constrains given in OntoCity (e.g., the misclassified area number 1 shown at the second row of Figure 4 cannot be wa-

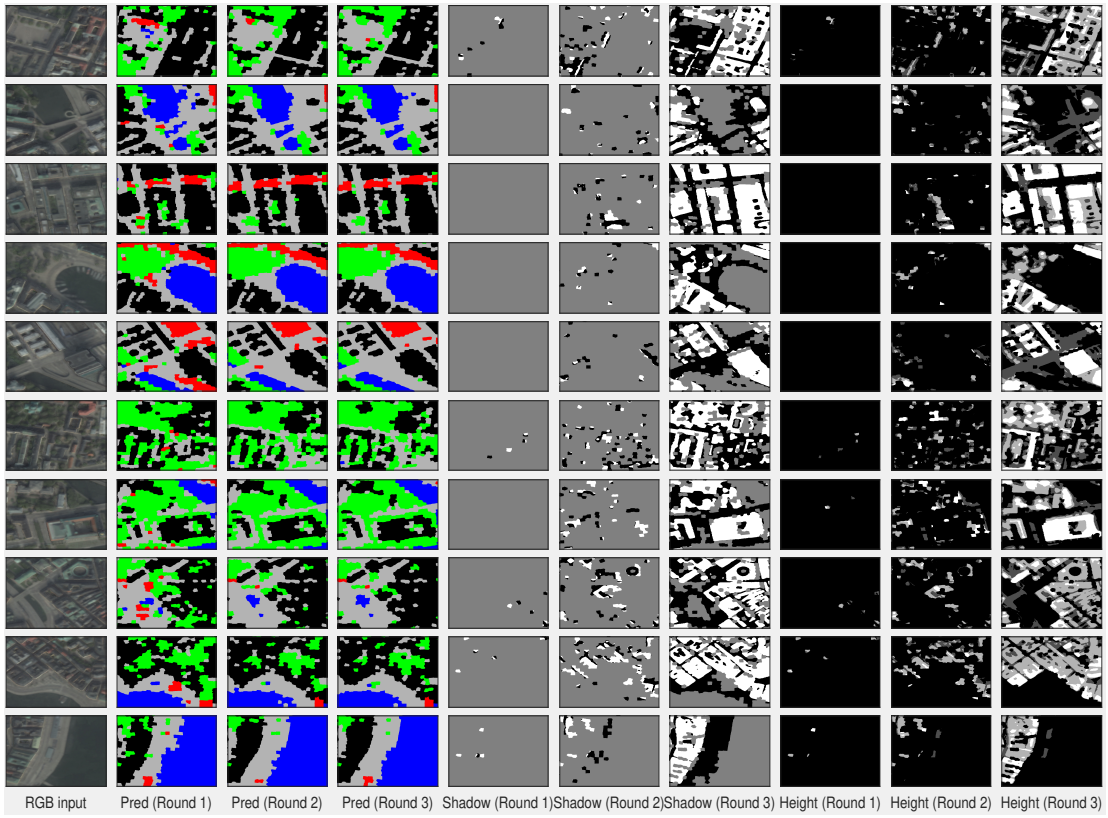

Fig. 5. RGB input (column 1), predictions from classifier for round 1 to 3 (column 2-4, green=vegetation, gray=road, black=building, blue=water, red=railroad), shadow estimations from reasoner for round 1 to 3 (column 5-7, gray=undefined, white=not shadow, black=shadow), and height estimations from reasoner for round 1 to 3 (column 8-10, black=low object, white=tall object).

ter nor railroad as the area is externally connected to a building.)

As an extra effort, we have applied the reasoner upon the misclassified areas to find their minimal set of possible labels w.r.t their neighborhood. Table. 3 shows the classification accuracy per class before and after the reasoner has adjusted the predictions from the classifier. The classes *vegetation* and *building* get an significant improvement while the other classes are rather unchanged.

	Vegetation	Road	Building	Water	Railroad
Before	78.0	76.7	88.2	97.2	97.7
After	90.4	77.7	95.9	97.2	97.7

Table 3

Classification accuracy (%) per class before and after the reasoner has adjusted the misclassifications from the predictions from the classifier in the last round.

5. Discussion & Future Work

In this work, we propose an ontology-based reasoning approach that improves the semantic segmentation of RGB satellite images where the classifier is able to learn from its mistakes by using a semantic referee. The semantic referee can explain the reason behind the misclassification based on spatial reasoning and the ontological knowledge about cities. Given the explanation about the errors, we have also proposed a method for correcting the errors.

It is worth mentioning that this work relies on the explanation that the referee suggests, which highly depends on the content of the available ontologies. Briefly speaking, the richer the ontological knowledge (in terms of spatial constraints), the more meaningful explanation we can expect from the reasoner. It is also worth to clarify that we do not categorize our current work as a neural-symbolic integrated system since the neural network algorithm is independent of the symbolic reasoning module which only interacts with the

classifier. We see this as a strength since this would allow different types of classifiers to be integrated with our system.

For future work, we envision the design of the semantic referee to be more integrated in the neural network such that the interaction between the two systems is not limited to only the first layer but instead part of the learning process of the hidden layers of the classifier as well. Another interesting future direction is to explore the reverse process, namely how the classifier can enhance the capabilities of the reasoner.

Acknowledgements

This work has been supported by the Swedish Knowledge Foundation under the research profile on Semantic Robots, contract number 20140033.

References

[1] M. Perumal, K.R. Sangeetha and B. Velumani, A Survey on Geographical Information System, Spatial Data mining and Ontology, *International Confernece on Intelligent Computing ApplicationsAt: Coimbatore* **1** (2014).

[2] L. Ma, M. Li, X. Ma, L. Cheng, P. Du and Y. Liu, A review of supervised object-based land-cover image classification, *ISPRS* **130** (2017), 277–293.

[3] G. Cheng, J. Han and X. Lu, Remote Sensing Image Scene Classification: Benchmark and State of the Art, *IEEE* (2017).

[4] J.E. Ball, D.T. Anderson and C.S. Chan, Comprehensive survey of deep learning in remote sensing: theories, tools, and challenges for the community, *Journal of Applied Remote Sensing* **11**(4) (2017), 042609.

[5] M. Zhang, X. Hu, L. Zhao, Y. Lv, M. Luo and S. Pang, Learning dual multi-scale manifold ranking for semantic segmentation of high-resolution images, *Remote Sensing* **9**(5) (2017), 500.

[6] E. Guirado, S. Tabik, D. Alcaraz-Segura, J. Cabello and F. Herrera, Deep-learning Versus OBIA for Scattered Shrub Detection with Google Earth Imagery: Ziziphus Lotus as Case Study, *Remote Sensing* **9**(12) (2017), 1220.

[7] A.G. Cohn, B. Bennett, J. Gooday and N.M. Gotts, Qualitative Spatial Representation and Reasoning with the Region Connection Calculus, in: *Proc. Dimacs Int. WS on Graph Drawing, 1994.*, 1997, pp. 89–4.

[8] M. Koubarakis, M. Karpathiotakis, K. Kyzirakos, C. Nikolaou and M. Sioutis, Data Models and Query Languages for Linked Geospatial Data, in: *Reasoning Web. Semantic Technologies for Advanced Query Answering - 8th International Summer School 2012, Vienna, Austria, September 3-8, 2012. Proceedings*, 2012, pp. 290–328.

[9] D.A. Randell, A. Galton, S. Fouad, H. Mehanna and G. Landini, Mereotopological Correction of Segmentation Errors in Histological Imaging, *J. Imaging* **3**(4) (2017), 63.

[10] A.S. Garcez, T.R. Besold, L.D. Raedt, P. Földiak, P. Hitzler, T. Icard, K. Kühnberger, L.C. Lamb, R. Miikkulainen and D.L. Silver, Neural-Symbolic Learning and Reasoning: Contributions and Challenges, in: *AAAI 2015 Spring Symposium on Knowledge Representation and Reasoning: Integrating Symbolic and Neural Approaches. T.R. SS-15-03*, 2015.

[11] J.M. Raymond, Integrating Abduction and Induction in Machine Learning, in: *Abduction and Induction*, Kluwer Academic Publishers, 2000, pp. 181–191. http://www.cs.utexas.edu/users/ai-lab/?mooney:bkchapter00.

[12] M. Alirezaie and A. Loutfi, Ontology Alignment for Classification of Low Level Sensor Data., in: *KEOD*, SciTePress, 2012, pp. 89–97. ISBN 978-989-8565-30-3.

[13] T.R. Besold, A.S. Garcez, S. Bader, H. Bowman, P. Domingos, P. Hitzler, K. Kuehnberger, L.C. Lamb, D. Lowd, P.M.V. Lima, L. de Penning, G. Pinkas, H. Poon and G. Zaverucha, Neural-Symbolic Learning and Reasoning: A Survey and Interpretation, *CoRR* **abs/1711.03902** (2017). http://arxiv.org/abs/1711.03902.

[14] S. Bader and P. Hitzler, Dimensions of Neural-symbolic Integration - A Structured Survey, *CoRR* **abs/cs/0511042** (2005). http://arxiv.org/abs/cs/0511042.

[15] D. Doran, S. Schulz and T.R. Besold, What Does Explainable AI Really Mean? A New Conceptualization of Perspectives (2017). http://arxiv.org/abs/1710.00794.

[16] N. Xie, K. Sarker, D. Doran, P. Hitzler and M. Raymer, Relating Input Concepts to Convolutional Neural Network Decisions (2017). https://arxiv.org/pdf/1711.08006.pdf.

[17] L.A. Hendricks, Z. Akata, M. Rohrbach, J. Donahue, B. Schiele and T. Darrell, Generating visual explanations, *Lect. Notes Comput. Sci.* **9908 LNCS** (2016), 3–19, ISSN 16113349. ISBN 9783319464923.

[18] J. Donahue, L.A. Hendricks, M. Rohrbach, S. Venugopalan, S. Guadarrama, K. Saenko and T. Darrell, Long-Term Recurrent Convolutional Networks for Visual Recognition and Description, *IEEE Trans. Pattern Anal. Mach. Intell.* **39**(4) (2017), 677–691. doi:10.1109/TPAMI.2016.2599174. https://doi.org/10.1109/TPAMI.2016.2599174.

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

[20] M.K. Sarker, N. Xie, D. Doran, M. Raymer and P. Hitzler, Explaining Trained Neural Networks with Semantic Web Technologies: First Steps, *CoRR* **abs/1710.04324** (2017). http://arxiv.org/abs/1710.04324.

[21] R.T. Icarte, J.A. Baier, C. Ruz and A. Soto, How a general-purpose commonsense ontology can improve performance of learning-based image retrieval, *IJCAI* (2017), 1283–1289, ISSN 10450823. ISBN 9780999241103.

[22] J. Masci, U. Meier, D. Cireşan and J. Schmidhuber, Stacked convolutional auto-encoders for hierarchical feature extraction, *Artificial Neural Networks and Machine Learning–ICANN 2011* (2011), 52–59.

[23] V. Nair and G.E. Hinton, Rectified linear units improve restricted boltzmann machines, in: *Proc. 27th Int. Conf. on machine learning (ICML-10)*, 2010, pp. 807–814.

[24] M.D. Zeiler, G.W. Taylor and R. Fergus, Adaptive deconvolutional networks for mid and high level feature learning, in: *Int. Conf. on Computer Vision (IEEE)*, 2011, pp. 2018–2025.

[25] X. Glorot and Y. Bengio, Understanding the difficulty of training deep feedforward neural networks, in: *Proc. 13th Int. Conf. on Artificial Intelligence and Statistics*, 2010, pp. 249–256.

[26] J. Duchi, E. Hazan and Y. Singer, Adaptive subgradient methods for online learning and stochastic optimization, *JMLR* **12**(Jul) (2011), 2121–2159.

[27] M. Alirezaie, A. Kiselev, M. Längkvist, F. Klügl and A. Loutfi, An Ontology-Based Reasoning Framework for Querying Satellite Images for Disaster Monitoring, *Sensors* **17**(11) (2017), 2545. doi:10.3390/s17112545. https://doi.org/10.3390/s17112545.

[28] M.A. Musen, The ProtÉGÉ Project: A Look Back and a Look Forward, *AI Matters* **1**(4) (2015), 4–12, ISSN 2372-3483. doi:10.1145/2757001.2757003. http://doi.acm.org/10.1145/2757001.2757003.

[29] F. Baader and W. Nutt, The Description Logic Handbook, Cambridge University Press, 2003, pp. 43–95, Chap. Basic Description Logics. ISBN 0-521-78176-0. http://dl.acm.org/citation.cfm?id=885746.885749.

[30] E. Sirin, B. Parsia, B.C. Grau, A. Kalyanpur and Y. Katz, Pellet: A Practical OWL-DL Reasoner, *Web Semant.* **5**(2) (2007), 51–53, ISSN 1570-8268. doi:10.1016/j.websem.2007.03.004. http://dx.doi.org/10.1016/j.websem.2007.03.004.

[31] A. Sanin, C. Sanderson and B.C. Lovell, Shadow Detection: A Survey and Comparative Evaluation of Recent Methods, *Pattern Recogn.* **45**(4) (2012), 1684–1695, ISSN 0031-3203. doi:10.1016/j.patcog.2011.10.001. http://dx.doi.org/10.1016/j.patcog.2011.10.001.