A Symbolic Approach for Explaining Errors in Image Classification Tasks

Marjan Alirezaie[1*], **Martin Längkvist**[1*], **Michael Sioutis**[1], **Amy Loutfi**[1],
[1] Center for Applied Autonomous Sensor Systems, Örebro University, Örebro, Sweden
firstname.lastname@oru.se

Abstract

Machine learning algorithms, despite their increasing success in handling object recognition tasks, still seldom perform without error. Often the process of understanding why the algorithm has failed is the task of the human who, using domain knowledge and contextual information, can discover systematic shortcomings in either the data or the algorithm. This paper presents an approach where the process of reasoning about errors emerging from a machine learning framework is automated using symbolic techniques. By utilizing spatial and geometrical reasoning between objects in a scene, the system is able to describe misclassified regions in relation to its context. The system is demonstrated in the remote sensing domain where objects and entities are detected in satellite images.

1 Introduction

Many machine learning algorithms are trained by optimizing a cost function that continuously measures the training errors during learning, and adapts the model parameters in order to minimize these errors. With this approach, the learning algorithms seem to 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 or describing why such errors have been made, but instead follows the defined rules for parameter updates given by the selected minimization optimization method.

For satellite image classification, a classifier that only uses 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 example, the class *water* looks similar to *shadows*, and buildings with gray roofs will look similar to roads in the RGB channels. One solution to this problem, that 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 [Ma *et al.*, 2017; Cheng *et al.*, 2017]. However, these works are impractical for satellite images that only contain RGB channels, such as Google Maps. Another possible solution to increase the performance is to change the architecture of the classifier in order to increase the capacity, e.g., by using deep Convolutional Neural Networks (DCNNs) [Ball *et al.*, 2017; Zhang *et al.*, 2017; Guirado *et al.*, 2017].

In this paper, instead of adding additional sources of information or experimenting with the architecture of the classifier, we aim to spatially explain the errors in terms of their structure and neighborhood. 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.

Integration of data-driven learning methods with symbolic reasoning has been identified in the literature as one of the key challenges in Artificial Intelligence [Garcez *et al.*, 2015]. Depending on the approaches to represent both low and high level data, such integration has been addressed under different names that include abduction-induction in learning [Raymond, 2000], structural alignment [Alirezaie and Loutfi, 2012], and neural-symbolic methods [Besold *et al.*, 2017; Bader and Hitzler, 2005]. With the increasing interest in connectionist learning systems, and in particular in deep learning methods, research on integrated neural-symbolic systems has recently made considerable progress. Such integrations are routinely referred to as explainable Artificial Intelligence (XAI), and used to provide better insights into the learning process [Doran *et al.*, 2017].

1.1 Related Work

As discussed in [Xie *et al.*, 2017], 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. The work presented in [Hendricks *et al.*, 2016] introduces a learning system based on a convolutional network LRCN [Donahue *et al.*, 2017] that provides

*Equal contribution

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, in this work, no specific symbolic representation was provided, and the features related to the objects are taken from the sentences already available for each image in the dataset (CUB dataset [Wah *et al.*, 2011]).

With focus on the knowledge model, the work presented in [Sarker *et al.*, 2017] 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)[1] 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.

Likewise, the work detailed in [Icarte *et al.*, 2017] relies on a general-purpose knowledge model, namely, the ConceptNet Ontology. In this work, 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. For this, 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 aforementioned 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.

1.2 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 and provide possible explanations which can later be used in correcting misclassification. Our work differentiates from the previous neural-symbol systems in two regards. Firstly, our method is able to find the most likely misclassified data (which can be rephrased as errors realization). 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.

The rest of the paper is structured as follows: In Section 2 we present the steps of our approach. Section 3 provides the technical details on the classification process. The symbolic module including the ontological knowledge model and the reasoning process is explained in Section 4. Our experimental evaluations are presented in Section 5, which is followed by a brief discussion on the future work in Section 6.

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

2 Approach

Figure 1 illustrates our approach of using background (ontological) knowledge to explain the errors from a classifier trained on satellite image data. The process is composed of several steps including: (1) error realization, (2) error characterization using geometrical/spatial reasoning, (3) error generalization based on the frequency, and (4) error explanation by aligning its features with the ontological knowledge (ontological reasoning). The inferred explanation can possibly contribute in the process of error correction (shown as dashed-line) and update the classification results.

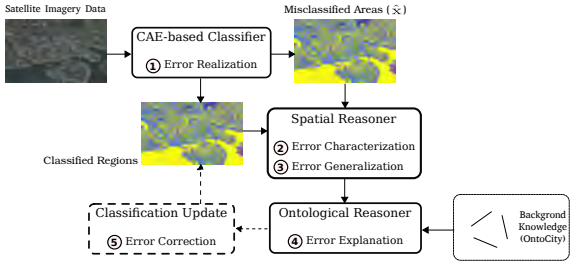

Figure 1: The process of explaining a misclassification in 4 steps: (1) error realization, (2) error characterization, (3) error generalization, and (4) error explanation. This process will contribute in error correction shown in dashed line.

Error realization refers to the process indicating likely misclassified areas (errors) on the map (to be detailed in Section 3). Given theses misclassified areas, a spatial/geometrical processing method characterizes such areas in terms of their structure and also identifies spatial relations within their vicinity. An ontological reasoning process is subsequently applied upon both the retrieved characterization of the errors and domain knowledge about generic spatial constraints in outdoor environments. After generalizing the relations retrieved by the reasoner based on their frequency, their semantics may justify the errors made by the classifier. Algorithm 1 provides further details of the explanation process.

Algorithm 1 Explaining Misclassification

Require: $S = empty, m, R$
▷ S: A hash-map, empty in the beginning
▷ m: The given misclassification matrix
▷ R: The given list of classified regions
$\mathcal{G} \leftarrow extractGeometries(m)$
for each $r \in \mathcal{R}$ **do**
  $t \leftarrow getRegionType(r)$
  for each $g \in \mathcal{G}$ **do**
    $q \leftarrow calculateRCC(g, r)$
    $S(q, t) \leftarrow < q, t >$
  end for
end for
$< Q, T > \leftarrow getMostSeenPair(S)$
$C \leftarrow queryOntology(Q, T)$
$Explanation \leftarrow getRegionType(C)$

The data used in this work consists of a RGB satellite image of central Stockholm, Sweden, with size 4000×8000

pixels and a pixel-resolution of 0.5 meters. The data was divided into patches of 500×500 pixels and divided into train and test sets by a $50 - 50$ split so that both sets contained a similar class distribution. The ground truth used in the classification process has been provided by the Swedish Mapping, Cadastral and Land Registration Authority (Lantmäteriet).

3 Object Detection and Classification

A Convolutional Auto-encoder (CAE) [Masci *et al.*, 2011] is used to classify every pixel in each sub-image of size 500×500 pixels into one of 5 categories, namely, vegetation, road, building, water, and railroad. One layer of a CAE consists of an encoder and a decoder, see Figure 2.

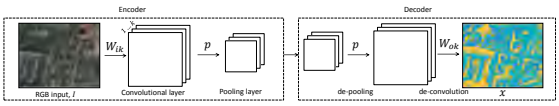

Figure 2: Overview of a one layer of Convolutional Autoencoder (CAE) that consists of an encoder and a decoder. The input is a RGB image and the output is the semantic segmentation.

The k-th feature map in the convolutional layer is calculated as:

$$h^k = \sigma_1 \left(I^i * W_{ik}^k + b^k \right) \tag{1}$$

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 that is applied to the whole map, σ_1 is a non-linear activation function, and $*$ denotes the convolution operation. In this work, we used the Rectified Linear Unit (ReLu) [Nair and Hinton, 2010] as the activation function. For an input image of size $m \times m \times c$ and a filter matrix of size $n \times n \times c \times k$, the convolutional layer is of size $(m-n+1) \times (m-n+1) \times k$.

The pooling layer is obtained by downsampling the convolutional layer by taking the maximum value in each $p \times p$ non-overlapping subregion. The size of the pooling layer is $(m - n + 1)/p \times (m - n + 1)/p \times k$.

The unpooling is performed with switch variables [Zeiler *et al.*, 2011] that remember the position of the maximum value during the pooling operation.

Finally, a deconvolutional operation is performed to obtain the final output, x. For a typical convolutional autoencoder, the output has the same dimensions as the input image, I. However, for our application we want to perform a classification of the input image. Therefore, the output image x has the dimensions $m \times m \times K$ where K is the number of classes. The K-th output layer denotes the probability of each pixel belonging to class K. The output layer is calculated as:

$$x = \sigma_2 \left(o^k * W_{ok}^k + c^K \right) \tag{2}$$

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, c^K is the bias for the K-th output layer, and σ_2 is the softmax non-linear activation function.

Figure 3 shows an overview of the method that is used to identify regions that have a high probability to be misclassified (i.e., the method to realize errors). The system consists of two Convolutional Auto-encoders (CAE) (noted *CAE 1* and *CAE 2* in Figure 3).

The first model, CAE 1, is trained to perform the image-to-image translation from the RGB input to the classified image x. CAE 1 is trained with supervised learning using the ground-truth y, see Section 3.

The second model, CAE 2, is trained unsupervised to reconstruct the input ground-truth y into the reconstruction of the ground-truth $\hat{y}$. The purpose of the model CAE 2 is to learn the overall structure and relation between classes.

The predicted label image x is then used as input to CAE 2 to get a reconstruction of the label image $\hat{x}$. The main idea is that regions that have a high reconstruction error, $(x - \hat{x})^2$, have a higher probability to be misclassified and should be further analyzed by the reasoner in order to explain a possible cause for the misclassification and give a suggestion for a more likely classification.

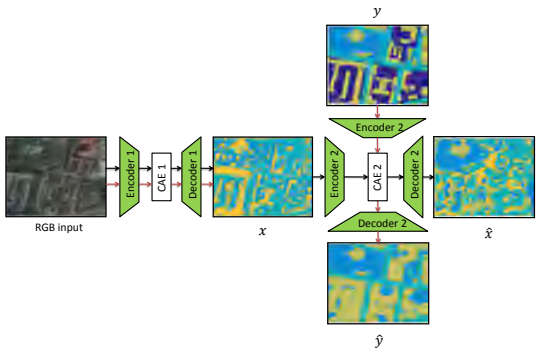

Figure 3: Overview of the method for indicating suspected misclassified regions. The input to the classifier (CAE 1) is an RGB image and produces the semantic segmentation, x. The label reconstructor (CAE 2) is first trained to reconstruct the groundtruth, y, into the reconstruction $\hat{y}$. The classified output x is then reconstructed using the label reconstructor to get the reconstructed classifications $\hat{x}$. The reconstruction error between x and $\hat{x}$ is then used to indicate the misclassified regions. Red arrows indicate the data processing during training and black arrows indicate the process during inference.

One important aspect of our method is the architecture for the label reconstructor in order to identify misclassified regions. On one hand, a single-layered CAE with a small filter size could easily reconstruct any configuration of the predicted map by simply reconstructing the local input pixel-by-pixel. Instead of increasing the filter size, we use a deep network with 5 layers. Due to the subsampling in each layer, this leads to the lower layers learn to reconstruct the local input and the higher layers learn the relation between areas with a larger perceptive field.

The classifier (CAE 1) and the label reconstructor (CAE 2) are constructed with the same architecture and consist of a 5-layer CAE. The filter size for each layer is $[11, 9, 7, 5, 3]$ and the number of filters in each layer is $[10, 20, 30, 40, 50]$. The pooling dimension is set to 2 in each layer and uses max-

pooling. The activation function in each layer is the ReLu-activation function except for last layer that uses a softmax activation function. The parameters were initialized with Xavier initialization [Glorot and Bengio, 2010] and trained using the AdaGrad [Duchi *et al.*, 2011] optimization method until convergence, which took around 50 hours on a GTX 1060 GPU.

4 Reasoning on misclassifications

The misclassification explanation process relies on geometrical and ontological reasoning. Before outlining the details of the explanation process, we first briefly introduce OntoCity which contains the background knowledge model used in this work.

4.1 OntoCity

OntoCity[2] is an extension of the GeoSPARQL[3] ontology that serves as a standard vocabulary for geospatial data by enabling qualitative spatial reasoning upon this type of data. OntoCity, whose representational details can also be found in [Alirezaie *et al.*, 2017], has been designed to represent cities in terms of different aspects including the structural details, conceptual and physical objects, their types (e.g., natural or man-made), and their relations (e.g., spatial constraints, affordances). The main concept in OntoCity is `oc:CityFeature`, which is the subclass of the class `geos:Feature`[4] and defines any spatial object with a geometry in the physical world. According to the following axiom[5], a city feature is (or more specifically is a subclass of) a feature whose geometry is in the form of a polygon and has at least one spatial relation with another city feature:

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

By spatial relation we refer to the 8 relations in RCC-8 (Region Connection Calculus) [Cohn *et al.*, 1997] that are also defined in GeoSPARQL, and are used to specialize the definition of features in a city. In OntoCity there are different types of features defined as subclasses of the `oc:CityFeature` class. For instance, a feature might be with a fixed geometry (`oc:FixedGeometryFeature`) or a dynamic one whose geometry changes in time (`oc:DynamicGeometryFeature`). Likewise, a feature can be physical (`oc:PhysicalFeature`, e.g., a landmark with absolute elevation value measured from the sea floor), conceptual (`oc:ConceptualFeature`, e.g., a rectangular division in a city regardless of their landmarks), mobile (`oc:MobileFeature`, e.g., a car), or stationary (`oc:StationaryFeature`, e.g., a building). The following axioms show some subsumption relations with `oc:CityFeature`:

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

Each of the aforementioned subclasses of the class `oc:CityFeature` has its own taxonomy. Due to the lack of space, we only mention a limited number of these axioms. For instance, `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 regions, or likewise, into man-made or natural ones:

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

A non-flat region in OntoCity refers to those landmarks with a relative elevation value, where by relative we mean the height measured from the ground level (in their neighborhood) and not from the absolute sea-level. Due to its height, a non-flat region is also assumed to cast shadows (defined as the class `oc:Shadow` in OntoCity) with which it has a spatial relation `oc:intersects` that subsumes several RCC-8 relations including partially overlapping (`geos:rcc8po`) and externally connected (`geos:rcc8ec`).

The subclasses of the class `oc:Region` can also specify the texture (i.e., type) of the landmark categorized as follows. in the following. It is worth mentioning that some of these region types are used as labels by the classifier to classify regions:

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

The RCC-8 relations are used in OntoCity to describe more specific features (e.g., bridges, shadows, shores) whose spatial relations with their vicinity are important in their definitions. For instance, a bridge is defined as a man-made region which is not flat (i.e, has elevation) and is partially overlapping (referring to the RCC-8 relation `geos:rcc8po`) at least another region (e.g., a water area, a street):

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

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

[2] https://w3id.org/ontocity/ontocity.owl

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

[4] The prefixes `oc` and `geos` refer to the URIs of OntoCity and GeoSPARQL, respectively.

[5] The axioms are in description logic (DL) [Baader and Nutt, 2003].

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 were a sub set of the general knowledge represented in OntoCity. However, the background knowledge can be much more specific and indicate unique features of a specific environment (e.g., "*in the given region there is no building connected to water areas*").

4.2 Explaining the misclassifications

The process of explaining the misclassifications is composed of several steps as shown in Algorithm 1. The algorithm accepts as input the list of the classified regions $\mathcal{R}$ as well as the misclassifications represented in the form of a pixel matrix $\mathtt{m}$ (as the reconstruction error between x and $\hat{x}$ explained in Section 3). In order to (spatially) characterize the errors, first the boundaries of the misclassified areas formed by misclassified pixels need to be extracted (see line 4 in the algorithm). Given the geometry of both the misclassified areas ($\mathcal{G}$) and the classified regions ($\mathcal{R}$) in the form of polygons, the algorithm calculates all the possible (RCC-8) qualitative spatial relations between any pairs of (g, r) where $g \in \mathcal{G}$ is a misclassified area and $r \in \mathcal{R}$ is a classified region in its vicinity. For each pair (g, r), besides the calculated spatial relation q, the algorithm also keeps the type of the region r shown 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 5-11). The information provided in S can be also seen as the geometrical characteristics of the misclassified areas (i.e., error characterization).

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. Let the pair $< Q, T >$ (see line 12) be the most observed spatial relation Q between the misclassified areas and a specific region type T, and let us view it as a representative feature of the misclassified areas. By applying an ontological reasoner upon OntoCity, we can query the ontology and ask for all the spatial features that are at least in one Q relation with type T, where the DL syntax of the query is: $\exists\ T.Q$. By applying the ontological reasoner the query can also be further generalized from the type T to its super-classes in OntoCity (see line 13). The concept (C) as a spatial feature ($C \sqsubseteq$ `oc:CityFeature`), which is inferred by the reasoner, is considered as an explanation.

5 Empirical Evaluation

The classifier was trained on the training set and applied upon the test data and resulted in $\approx 32K$ segments (or regions). Figure 4, left column, shows a 500×500 pixel large subset of the test data together with the segmentation. Each segment is classified into vegetation, road, building, water, or railroad (middle column). The reconstruction error (right column) identifies the probability that the segment is misclassified, in particular, the darker the segment the less likely it is to be misclassified (i.e., error realization).

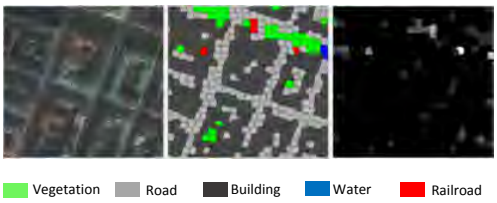

Figure 4: Left: Input RGB image together with the segmentation. Middle: Classified segmentation from the classifier. Right: Average reconstruction error for each segment where bright areas indicate suspected classification errors.

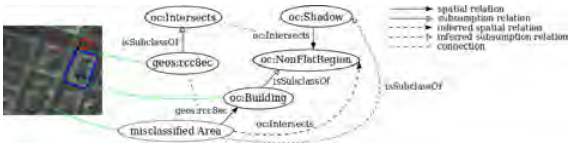

Figure 5: A high level representation of an example error explanation process. The *misclassified area* (in red) is *externally connected* (`geos:rcc8ec`) to the *building* region (in blue). By mapping the 3 aforementioned entities into their equivalent concepts in the ontology, the ontological reasoner infers the direct superclass (i.e., `oc:shadow`) of the misclassified area whose constraints are more general ($\exists$ `oc:intersects.oc:NonFlatRegion`) than the spatial representation of the red misclassified area.

Given the segments and the sorted list of reconstruction errors, the spatial reasoner together with the ontological reasoner are in charge of error explanation. The high level representation of the symbolic process is illustrated in Figure 5. In the following we go through the details of each step requited to achieve the final explanations for the errors. The error characterization process as the first step considers the top 100 misclassified regions to extract their boundaries and their spatial relations with their segmented neighborhood. This step has been implemented using the open-source JTS Topology Suite[6]. Table 1 shows a summary of the error characterization process. To find a representative feature of the misclassified areas (i.e., error generalization), Algorithm 1 takes into account 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 89 misclassified areas.

The pair $< Q, T >$ is enough to query the ontological concepts with spatial constraints. We have extended and used the reasoner Pellet, as an open-source Java based

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

Type (t) \ Relation (q)	ec	po
oc:Building	**89**	5
oc:Road	41	0
oc:Water	19	1

Table 1: A summary on the error characterization process: 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.

OWL 2 ontological reasoner [Sirin *et al.*, 2007]. The extension is in terms of filtering concepts based on their spatial constraints. The DL syntax of the query given to the reasoner is $\exists$ geos:rcc8ec.oc:Building interpreted as "*all the things 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. As shown in Table 2, 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 direct generalization of the query $\exists$ geos:rcc8ec.oc:Building (where, geos:rcc8ec $\sqsubseteq$ oc:intersects and oc:Building $\sqsubseteq$ oc:NonFlatRegion) (see Figure 5).

Inferred concepts by the reasoner	description
oc:CityFeature	indirect superclass
oc:ConceptualFeature	indirect superclass
oc:DynamicGeometryFeature	indirect superclass
oc:MobileFeature	indirect superclass
oc:Shadow	**direct** superclass

Table 2: Error explanation as the output of the ontological reasoner.

Figure 6 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 2, 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, 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.

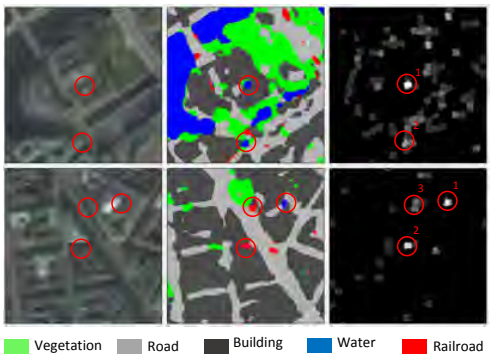

Figure 6: Two examples of the classification output along with their input RGB image, classified segmentation and the average reconstruction error. The misclassified areas marked with numbers are in spatial relations with buildings, roads, vegetation, etc. The ontological reasoner explains the misclassification as the result of the shadow of buildings on their neighborhood.

6 Discussion & Future Work

In this paper, we have proposed an ontology-based reasoning approach that automates the process of making sense of the misclassifications. The symbolic module (i.e., the spatial and ontological reasoning) used in this approach can act as a referee who explains why something has been misclassified. This explanation is made based on the geometrical features of the data which are not used by the classifier that only relies on the RGB channels of satellite image data.

Given the explanation about the errors, we ideally would like the symbolic module to also provide a correction of the misclassifications (see Figure 1). For this, there are a number of issues that have not been addressed in this work. The correction process depends on the inferred concept from the ontology. For example, if the concept as the explanation refers to a specific region type (i.e, a physical concept such as oc:Bridge) we could relabel the misclassified pixels with the region type. However, as we have shown, it can be a conceptual feature for which finding a relevant label to relabel the misclassified pixels might need further processing. If the reasoner infers that the misclassified area is under shadow, for example, the new label for this area is assumed to be the same as the type of the regions surrounding (referring to the RCC-8 relation *tangential proper part*: geos:rcc8tpp) the area under shadow. As the next step, we will focus on the correction process and deal with the aforementioned issues.

Acknowledgments

This work has been supported by the Swedish Knowledge Foundation under the research profile on Semantic Robots, contract number 20140033. The authors would also like to thank Mehul Bhatt at the Center for Applied Autonomous Sensor Systems for useful discussions that contributed to this paper.

References

[Alirezaie and Loutfi, 2012] M. Alirezaie and A Loutfi. Ontology alignment for classification of low level sensor data. In *KEOD*, pages 89–97. SciTePress, 2012.

[Alirezaie *et al.*, 2017] 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):2545, 2017.

[Baader and Nutt, 2003] F. Baader and W. Nutt. The description logic handbook. chapter Basic Description Logics, pages 43–95. Cambridge University Press, 2003.

[Bader and Hitzler, 2005] S. Bader and P. Hitzler. Dimensions of neural-symbolic integration - A structured survey. *CoRR*, abs/cs/0511042, 2005.

[Ball *et al.*, 2017] 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):042609, 2017.

[Besold *et al.*, 2017] 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.

[Cheng *et al.*, 2017] G. Cheng, J. Han, and X. Lu. Remote sensing image scene classification: Benchmark and state of the art. *IEEE*, 2017.

[Cohn *et al.*, 1997] 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.*, pages 89–4, 1997.

[Donahue *et al.*, 2017] 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):677–691, 2017.

[Doran *et al.*, 2017] D. Doran, S. Schulz, and T.R. Besold. What Does Explainable AI Really Mean? A New Conceptualization of Perspectives. 2017.

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

[Garcez *et al.*, 2015] 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.

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

[Guirado *et al.*, 2017] 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):1220, 2017.

[Hendricks *et al.*, 2016] L.A. Hendricks, Z. Akata, M. Rohrbach, J. Donahue, B. Schiele, and T. Darrell. Generating visual explanations. *Lect. Notes Comput. Sci.*, 9908 LNCS:3–19, 2016.

[Icarte *et al.*, 2017] 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*, pages 1283–1289, 2017.

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

[Masci *et al.*, 2011] 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*, pages 52–59, 2011.

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

[Raymond, 2000] J.M. Raymond. Integrating abduction and induction in machine learning. In *Abduction and Induction*, pages 181–191. Kluwer Academic Publishers, 2000.

[Sarker *et al.*, 2017] Md. 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.

[Sirin *et al.*, 2007] E. Sirin, B. Parsia, B.C. Grau, A. Kalyanpur, and Y. Katz. Pellet: A practical owl-dl reasoner. *Web Semant.*, 5(2):51–53, June 2007.

[Wah *et al.*, 2011] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD Birds-200-2011 Dataset. Technical report, 2011.

[Xie *et al.*, 2017] N. Xie, K. Sarker, D. Doran, P. Hitzler, and M. Raymer. Relating Input Concepts to Convolutional Neural Network Decisions. (2016), 2017.

[Zeiler *et al.*, 2011] 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)*, pages 2018–2025, 2011.

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