Semantic Referee: A Neural-Symbolic Framework for Enhancing Geospatial Semantic Segmentation

Marjan Alirezaie a,*, Martin Längkvist a, Michael Sioutis a and Amy Loutfi a

a Center for Applied Autonomous Sensor Systems, Örebro University, Örebro, Sweden

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 semantic referee, which is able to extract qualitative features of 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 reasoner outputs a semantic augmentation for the errors that is then reported back to the learning algorithm to learn from its mistakes. In this paper, the proposed method of the interaction between a neural network classifier and a semantic referee shows how to improve the performance of semantic segmentation for satellite imagery data.

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

1. Introduction

Machine learning algorithms and semantic web technologies have both been widely used in geographic information systems [1], [2]. The former are typically applied on geo-data to perform image recognition tasks including object recognition, whereas the latter are used for a number of applications such as navigation, knowledge acquisition and map query [3]. Despite recent success in machine learning, in particular, with deep learning methods for image segmentation and classification of satellite data, seldom do these approaches take into account the advantages of the semantics that are associated with geo-data. Instead, training processes in machine learning algorithms typically rely on optimizing a cost function that measures the errors during learning and adapt the model parameters to minimize these errors. Seemingly, these algorithms learn from their errors; however, it does not mean that they are able to know why such errors occurred or conceptualize them using their features. In other words, the training process of minimizing a cost function only follows some rules for parameter updates that are predefined by the selected optimization method, and does nothing towards describing why such errors have been made.

In the context of semantic segmentation and classification for geospatial data, a classifier that accepts
as input only the RGB channels is error-prone due to the visual similarity between certain classes of data. For example, in satellite imagery, the RGB channel water looks similar to shadows, and buildings with gray roofs look similar to roads. One solution to this problem studied in the literature, is to include additional sources of information as part of the input data to the classifier. Examples of these extra sources include 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 [4, 5]. However, such data is not always accessible, as is the case with satellite images from Google Maps that only contain RGB channels and other similar data 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) [6–8].

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 conceptualizing the errors in terms of their spatial relations and neighborhood instead, as a means to ameliorate the performance of machine learning algorithms. For this to be possible, we propose to apply a reasoner upon an ontological representation of the context in order to retrieve the spatial and geometrical characteristics of the data. We refer to this process as a semantic referee, because we use knowledge representation and reasoning methods 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 [9], 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 standard, and has found its way into various Semantic Web tools and applications over the past few years [10]. A worthwhile 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 [11]. In our work, inspired by the integration of qualitative spatial reasoning methods into imaging procedures as described in [11] (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.

One of the key challenges in Artificial Intelligence is about reconciliation of data-driven learning methods with symbolic reasoning [12]. The integration approaches between low and high level data have been addressed under different names depending on the employed representational models and include abduction-induction in learning [13], structural alignment [14], and neural-symbolic methods [15, 16]. Due to the increasing interest in deep learning methods, design and development of neural-symbolic systems has recently become the focus of different communities in artificial intelligence, as they are assumed to provide better insights into the learning process [17].

1.1. Contribution

In this work, we develop an ontology-based reasoning approach, a preliminary version of which can be found in [18], to assist a neural network classifier for a semantic segmentation task. This assistance can be used in particular to represent typical errors and extract their features that eventually assist in correcting misclassification. We show using a specific case on large scale satellite data how semantic web resources interacts with deep learning model to improve the classification performance 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 conceptualize 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 its features. Secondly, our model focuses on the misclassifications and uses ontological knowledge together with a geometrical processing to explain them. This combination, to the best of our knowledge, is the first time to be employed for the aforementioned purpose. 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.

1http://www.opengeospatial.org/standards/geosparql
1.2. Structure of paper

The rest of the paper is structured as follows. Section 2 describes the related work. The method is presented 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 conceptualization process and how it 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 [19], 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 [20] introduces a learning system based on a Long-term Convolutional Network (LTCN) [21] 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 [22]).

With focus on the knowledge model, the work presented in [23] 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 [24] 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, such as their affordances and their relations with other objects, is aligned with objects derived from the deep learning method.

The method of enriching the data by providing information as additional channels for training a CNN-based network has been done before. The works of [25] and [26] augment the input data by adding two additional channels that represent the \(i\) and \(j\) coordinates in the image to obtain the location information. Our work uses information from a semantic referee as the augmented data instead of the location information.

Although in 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.

Our approach can also be compared with Explanation-based learning (EBL) [27] approaches. EBL refers to a form of machine learning method that is able to learn by generalizing examples where the features of the examples are formalized as domain theory. In EBL, the explanations which consist of the features of the observation is directly considered and generalized by the learner, whereas in our semantic based model, although the features of the misclassified areas are inferred from the ontology and send back to the classifier, they are not directly applied on the classification output; rather, they are only be treated as a new set of data that sent through the learning process. It means that the reasoner feedback is not necessarily and directly is taken into account on the classification outputs.

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 seman-
In order to deal with the misclassifications (errors) made by the classifier, we developed 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 within the learning process.

### 3.2. Data

The data used in this work consists of RGB satellite images from two different cities in Sweden, shown in Figure 2. The first city is Stockholm, which is the largest city and capital of Sweden and the second city is a smaller city located in northern Sweden called Boden. The selected area size for both cities is $4000 \times 8000$ pixels with a pixel-resolution of 0.5 meters and was divided into train and test sets with a 50 – 50 split. The ground 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 are vegetation, road, building, water, and railroad. The class distribution for each city can be seen in Table 1. Due to the large imbalance in the data set, the loss function uses median frequency class weighting.

---

3. https://www.lantmateriet.se/
Table 1

<table>
<thead>
<tr>
<th></th>
<th>Vegetation</th>
<th>Road</th>
<th>Building</th>
<th>Water</th>
<th>Railroad</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stockholm (train)</td>
<td>7.6</td>
<td>31.3</td>
<td>35.4</td>
<td>23.5</td>
<td>2.2</td>
</tr>
<tr>
<td>Stockholm (test)</td>
<td>18.2</td>
<td>36.9</td>
<td>19.7</td>
<td>22.4</td>
<td>2.8</td>
</tr>
<tr>
<td>Boden (train)</td>
<td>63.0</td>
<td>19.7</td>
<td>4.9</td>
<td>11.0</td>
<td>1.3</td>
</tr>
<tr>
<td>Boden (test)</td>
<td>54.5</td>
<td>25.6</td>
<td>10.8</td>
<td>7.2</td>
<td>1.8</td>
</tr>
</tbody>
</table>

Class distribution for the two cities used in this work. There is large difference in amount of vegetation, building, and water between the two cities. Both cities have a very small amount of railroads.

Fig. 2. The data consists of RGB satellite images from two different cities Stockholm and Boden. The selected area size for both cities 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.

3.3. Data classification

A variation of a Convolutional Auto-encoder (CAE) [28] is used to perform the semantic segmentation of the satellite images where every pixel in the map is classified. The structure of the networks follows the model U-net [29] and is created in MATLAB 2018a with the function creatUnet with patchsize 256, 6 color channels (3 RGB channels + 3 channels as the feedback provided by the semantic referee (see Figure. 1)), and 5 classes shown in Table 1.

The model consists of an encoder with 4 layers where each layer performs two convolutions with 64×$L$ filters in the $L$th layer and a 2 × 2 max-pooling operation; a middle section that performs two convolutions with 1024 filters and 50% dropout; and a decoder with 4 layers that performs deconvolution, depth concatenation with the output from the encoder at the same layer, and two convolutions. The number of filters in each layer $L$ in the decoder is 512/[$L$]. Each convolution and deconvolution is followed by a ReLu-activation [30] and has a filter size of $3 \times 3$ and $2 \times 2$, respectively. The classification of each pixel is performed with a convolution with 5 filters with filter size $1 \times 1$ followed by a softmax activation function for the final per-pixel classification.

The model parameters are trained from scratch and were initialized with Xavier initialization [31] and trained using the Adam optimization method [32] with initial learning rate $10^{-4}$ and minibatch size 20 with early-stopping using a validation set that was randomly drawn from 10% of the training data.

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 and contains the domain knowledge about generic spatial constraints in outdoor environments. OntoCity whose (part of) representational details can also be found in [33] is an extension of the GeoSPARQL ontology, known as a standard vocabulary for geospatial data [10]. The main idea behind designing OntoCity was to develop a generalized knowledge model to represent cities in terms of their structural, conceptual and physical aspects as well as the types of these aspects (e.g., natural or man-made) and their relations (e.g., spatial constraints, affordances). Figure 3 illustrates a Protégé [34] snapshot of the hierarchy of concepts defined in OntoCity.

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 contains a prefix that indicates the ontology to which it belongs. In the aforementioned classes, the two prefixes $oc$ and $geos$ refer to the URIs (Uniform Resource Identifiers) of the two ontologies OntoCity and GeoSPARQL, respectively.

https://w3id.org/ontocity/ontocity.owl
Fig. 3. A snapshot of the hierarchy of concepts in OntoCity. The city features are defined as the subclasses of the geos:Feature class defined in GeoSPARQL.

The class geos:Feature, which represents any spatial object with some geometry, subsumes the class oc:CityFeature, which represents features in a city in the form of polygons and that share at least one spatial relation with the remaining city features. The axioms of OntoCity given in this paper are in description logic (DL) [35]:

\[
\text{oc:CityFeature} \sqsubseteq \text{geos:Feature} \sqcap \\
\exists \text{geos:hasGeometry}.\text{geos:Polygon} \sqcap \\
\exists \text{oc:hasSpatialRelation.}\text{oc:CityFeature}
\]

Spatial relations in OntoCity include the RCC-8 (Region Connection Calculus) [9] relations defined in [9] 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 spatial relation oc:intersects is used to simplify the representation of some situations for which we only need to know whether the two features are intersecting or not.

Spatial relations are used in the form of spatial constraints to provide meaning to the city features. City features are categorized into several types defined as the subclasses of oc:CityFeature in OntoCity. These categories include oc:PhysicalFeature and oc:ConceptualFeature, which represent 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 represent features whose geometries are fixed or dynamic (changing in time) respectively. Mobility is another property that categorizes the city features into mobile (oc:MobileFeature, e.g., a car), or stationary (oc:StationaryFeature, e.g., a building). The following axioms show the aforementioned subsumption relations:

\[
\text{oc:DynamicGeometryFeature} \sqsubseteq \text{oc:CityFeature} \\
\text{oc:FixedGeometryFeature} \sqsubseteq \text{oc:CityFeature} \\
\text{oc:MobileFeature} \sqsubseteq \text{oc:CityFeature} \\
\text{oc:StationaryFeature} \sqsubseteq \text{oc:CityFeature} \\
\text{oc:ConceptualFeature} \sqsubseteq \text{oc:CityFeature} \\
\text{oc:PhysicalFeature} \sqsubseteq \text{oc:CityFeature} \sqcap \\
\]
∃ 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 that is also stationary (i.e., non-mobile) represents a landmark that can be categorized into various types such as flat or non-flat, or likewise, into man-made or natural regions:

```
oc:Region ⊑ oc:PhysicalFeature ⊓ oc:StationaryFeature
oc:ManmadeRegion ⊑ oc:Region
oc:NaturalRegion ⊑ oc:Region
oc:FlatRegion ⊑ oc:Region
oc:NonFlatRegion ⊑ oc:Region ⊓ ∃ oc:hasRelativeElevationValue.xsd:double ⊓ ∃ oc:intersects.oc:Shadow
```

For each location in a city (or in general on the ground) there are two elevation values, 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 height 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 of 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 (i.e., classes listed in Table 1) taken into account by the classifier to classify regions. These regions are defined as follows:

```
oc:River ⊑ oc:WaterArea ⊑ oc:Region
oc:Road ⊑ oc:PavedArea ⊑ oc:ManmadeRegion
oc:Park ⊑ oc:VegetationArea ⊑ oc:Region
oc:Building ⊑ oc:ManmadeRegion ⊓ 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 that is partially overlapping (referring to the RCC-8 relation geos:rcc8po) at least one other region, the texture of which identifies 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:

```
oc:Bridge ⊑ oc:ManmadeRegion ⊓ oc:NonFlatRegion ⊓ ∃ geos:rcc8po.oc:Region
```

As one of the non-physical (conceptual) features defined in OntoCity, we can refer to the concept of shadow as a spatial feature with a dynamic and also mobile geometry (i.e., changing depending on the time of the day). Although the exact shape of shadows and their exact positions depend on many quantitative parameters including the position of the source light and the height value of the casting objects, it is still possible to qualitatively describe shadows in the ontology. The definition of the concept shadow in OntoCity is more precise because it also contains a spatial constraints saying that, for the concept to be a shadow, it needs to be intersecting (oc:intersects) with at least one non-flat region (likely as its casting object):

```
```

3.4.1. Specialization of OntoCity

The OntoCity axioms mentioned in the previous sections are a subset of general knowledge that always hold regardless of the city under study (e.g., “Water bridges cross water areas”). However, depending on the case study, the background knowledge might be specialized to represent features belonging to a specific environment (e.g., “in the given region there is no building connected to water areas”).

The areas under our study, as shown in Section 3.2, comprise the central part of Stockholm and also another small city Boden in north of Sweden. The following spatial constraints are valid for both of these cities and that is why have they been added to the version of OntoCity used in our case:

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 railroad tracks (referring to the negation of the \texttt{oc:intersects} relation).
3. Buildings are not directly connected to water-areas (referring to the negation of externally connected relation in RCC8: \texttt{geos:rcc8ec}).
4. Buildings are not directly connected to rail roads (referring to the negation of externally connected relation in RCC8: \texttt{geos:rcc8ec}).
5. Buildings are not contained by roads (referring to the negation of tangential proper part relation in RCC-8: \texttt{geos:rcc8tpp}).
6. Buildings do not contain roads (referring to the negation of tangential proper part inverse relation in RCC-8: \texttt{geos:rcc8tppi}).
7. Railroads are not directly connected to water-areas (referring to the negation of the \texttt{oc:intersects} relation).

The output of the classifier is in the form of labeled classes that can be used by a geometric algorithm to detect misclassified areas and augment the classification. The algorithm calculates all the possible (RCC-8) qualitative spatial relations between any pairs of classified regions (quantifying the certainty probability). The regions with low certainty are suspected to be misclassified ones and should be prioritized for inspection by the reasoner.

Given the output of the classification, which includes both the classified and misclassified regions for each segment, the (spatial/ontological) reasoner as a semantic referee is able to first conceptually generalize the errors and then semantically augment 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 Semantic Augmentation

Require: \( W = \text{empty}, S, m, R \)
1: \( \triangleright W: \text{A hash-map, empty in the beginning} \)
2: \( \triangleright S: \text{The given list of rectangular segments} \)
3: \( \triangleright P: \text{The given list of misclassified areas} \)
4: \( \triangleright R: \text{The given list of classified regions} \)
5: \( \text{for each } s \in S \text{ do} \)
6: \( P_s \leftarrow \text{getRegionsInSegment}(P, s) \)
7: \( R_s \leftarrow \text{getRegionsInSegment}(R, s) \)
8: \( \text{for each } r \in R, \text{ do} \)
9: \( t \leftarrow \text{getRegionType}(r) \)
10: \( \text{for each } p \in P, \text{ do} \)
11: \( q \leftarrow \text{calculateRCC}(p, r) \)
12: \( W\text{add}(<q, t>) \)
13: \( \text{end for} \)
14: \( \text{end for} \)
15: \( \text{end for} \)
16: \( <Q, T> \leftarrow \text{getHighFrequentSpatialRelations}(W) \)
17: \( C \leftarrow \text{queryOntology}(Q, T) \)
18: \( \text{augmentation} \leftarrow \text{getSemantics}(C) \)
```

The algorithm accepts as input the list of segments \( S \) and the list of regions in the form of polygons for both classified \( R \) and misclassified areas \( P \). For each segment, the algorithm extracts all the classified \( R_s \) and misclassified regions \( P_s \) that belong to the segment. Given the two lists of polygons \( R_s \) and \( P_s \), the algorithm calculates all the possible (RCC-8) qualitative spatial relations between any pairs of \( (p, r) \) where \( p \in P_s \) is a misclassified area and \( r \in R_s \) is a classified region in its vicinity.

For each pair \( (p, r) \), the algorithm calculates the spatial relation \( q \) between \( p \) and \( r \) and also keeps the type of the region \( r \) named as \( t \). All the calculates pairs \(<q, t>\) are added to the list \( W \). The list \( W \) will at the end contain all the spatial relations that exist between the misclassified areas for each specific region type (see lines 5-15). In other words, \( W \) is defined to
contain the geometrical characteristics of the misclassified areas.

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 16) 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 queries OntoCity to find all the spatial features that are at least in one \(Q\) relation with the region 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 17). The concept \((C)\) as a spatial feature \((C \sqsubseteq \text{oc:CityFeature})\) inferred by the reasoner, is considered as the semantic augmentation for the misclassified regions that are in the given spatial relations with the given region type.

The computational complexity of the algorithm has the order of magnitude \(O(k \times m \times n)\), where \(k\) is the number of segments, \(m\) is the number of classified and \(m\) is the average number of misclassified regions in the segment.

3.6. Closing the loop

There are a number of ways that the output from the reasoner (i.e., the semantically augmented errors) can influence a neural network-based classifier, e.g., training set selection, data selection, architecture design, and cost function modification. This work uses a limited amount of training data with only RGB channels as input so there are few options for the reasoner to perform training set and data selection. This work also uses a standard model, namely U-Net with class weighing as cost function modification. Therefore, in this work, the reasoner will influence the classifier by providing additional information that is generated by the reasoner as a way to augment the original training data. This information is represented as input channels to the classifier in addition to the standard RGB channels. The information that the reasoner provides to the classifier highly depends on the semantics of the inferred semantic augmentations. As we will see in Section 4, the reasoner finds shadows as one major cause behind many of the misclassifications. 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 is there any formal representation to calculate the location of shadows, this explanation as a semantic referee provides a significant insight for us to develop the geometrical reasoner to localize the shadows.

Furthermore, the ontological reasoner finds many suspicious areas whose spatial relations with their neighborhood were inconsistent and violating the constraints defined in OntoCity (see Section 3.4.1).

The information about the regions that are inferred to be under shadows, along with the other inconsistent regions, are sent back to the reasoner in the form of channels of data. Each data channel is represented by a matrix of pixels with the same size as the input data. Further details are given in Section 4.

4. Empirical Evaluation

4.1. Error Characterization

Since the ground truth is available for our data, it is possible to calculate the certainty of misclassified regions. In this work, we use the classification certainty and select all the regions whose classification certainty is less than 70% and consider them as (likely) misclassified regions. Given both the classified regions and the misclassified areas, as explained in Section 3.5, the spatial reasoner together with the ontological knowledge are able to conceptualize the errors. The conceptualization process is based on extracting the spatial relations of the misclassified areas with their segmented neighborhood. This step has been implemented using the open-source JTS Topology Suite\(^3\), whose summary of results for Stockholm and Boden are shown in Table 2 and Table 3. Each cell of the table represents number of misclassified areas that are in a spatial relation (given in the column header) with all the regions with a specific type (given in the row header).

Given the Stockholm test data classification outputs, the reasoner, in order to find a representative feature of the misclassified areas, considers the pair \(<Q,T>\) as the most observed spatial relation \(Q\) between the misclassified areas and a specific region type \(T\). Table 2 shows the results of the first round of the classification. As we can see, the most observed spatial relations that involves 136 misclassified areas is the pair \(<Q=\text{geos:rcc8ec}, T=\text{oc:Building}>\).

\(^3\)https://github.com/locationtech/jts
Given the pair \( < Q, T > \), the reasoner queries the ontology with spatial constraints. The ontological reasoner that we have used in this work is the extended version of the reasoner Pellet, as an open-source Java based OWL 2 ontological reasoner \[36\]. 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 \( \exists \text{geos:}\text{rcc8ec}.\text{oc:}\text{Building} \) interpreted as “all the entities that are at least in one \text{geos:}\text{rcc8ec} relation with the region type \text{oc:}\text{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 \text{oc:}\text{shadow} (as a direct answer of the query). In OntoCity, the concept shadow is defined based on the spatial constraint: \( \exists \text{oc:}\text{intersects}.\text{oc:}\text{NonFlatRegion} \), which is found by the reasoner as the generalization of the query \( \exists \text{geos:}\text{rcc8sec}.\text{oc:}\text{Building} \) (where \( \text{geos:}\text{rcc8sec} \subseteq \text{oc:}\text{intersects} \) and \( \text{oc:}\text{Building} \subseteq \text{oc:}\text{NonFlatRegion} \)) (see Figure 1, top layer).

Table 2

<table>
<thead>
<tr>
<th>Type (t)</th>
<th>Relation (q)</th>
<th>ec</th>
<th>po</th>
</tr>
</thead>
<tbody>
<tr>
<td>oc:Building</td>
<td>ec</td>
<td>136</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>po</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>oc:Water</td>
<td>ec</td>
<td>11</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>po</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

Summary of the inconsistent spatial features of errors in classification of Stockholm test data. Each cell value represents the number of misclassified regions involved in the given spatial relations with the given region type, where \text{ec} and \text{po} refer to the RCC-8 relations \text{externally connected} and \text{partially overlapping}, respectively.

Table 3

<table>
<thead>
<tr>
<th>Type (t)</th>
<th>Relation (q)</th>
<th>ec</th>
<th>nttpi</th>
</tr>
</thead>
<tbody>
<tr>
<td>oc:Building</td>
<td>ec</td>
<td>164</td>
<td>118</td>
</tr>
<tr>
<td></td>
<td>nttpi</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>oc:RailRoad</td>
<td>ec</td>
<td>93</td>
<td>0</td>
</tr>
</tbody>
</table>

Summary of the inconsistent spatial features in classification of Boden test data. Each cell value represents the number of misclassified regions involved in the given spatial relations with the given region type, where \text{ec} and \text{nttpi} refer to the RCC-8 relations \text{externally connected} and \text{non-tangential proper part inverse} (i.e., \text{ogc:}\text{contains}), respectively.

Figure 4 illustrates two samples taken from Stockholm test set classification output where the misclassified areas are marked in red. At the first row, the areas marked with number 1 and 2 are misclassified as water. As the RGB image on the left shows, the misclassified areas (in red) are (externally) connected to buildings that cast shadows. At the second row, the area marked with number 1 is likewise misclassified as water. This area is again (externally) connected to a building. This area is also located between (i.e., connected with) at least two disconnected regions labeled as roads that are disconnected at the shadow area. This combination can explain the second most observed relation listed in Table 2, between the misclassified areas and the region type \text{oc:}\text{Road}. Assuming that buildings are often located close to roads (or streets), their shadow is 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, but 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.
consistent w.r.t. the constraints defined in OntoCity (see Section 3.4.1). As shown in Table 3, 93 rail roads were connected to other (misclassified) regions (e.g., buildings and water areas), a fact which according to OntoCity is inconsistent. Likewise, 164 buildings are connected to other regions, 67 cases out of which were again according to OntoCity inconsistent, and the remaining 97 cases (i.e., the consistent ones) were inferred as shadows. Moreover, 118 misclassified regions (mainly roads) were spatially contained in buildings. However, the ontological reasoner found them inconsistent as according to OntoCity roads cannot be contained (surrounded) by buildings.

4.2. Feedback to the Classifier

Knowing the main reason behind the misclassification, the semantic referee is expected to guide the classifier to better tackle its mistakes. For this, the semantic augmentation of misclassified regions as the output of the reasoner, which is either in the form of a new inferred concept (e.g., shadow) or suspicious regions (i.e., the inconsistent region types), is sent back to the classifier. One of the strategies for representing the reasoner’s feedback to the classifier is to provide more channels of data. We have defined three channels.

The first channel belongs to the shadow pixels. There is a fair amount of research work with the focus on shadow detection in the fields of computer vision and pattern recognition [37]. However, since the focus of this work is not to design an algorithm for shadow detection, and we are interested in informing the classifier of possible causes for the misclassification instead, 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 (second channel). 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 the 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.

Finally, the third channel of data is dedicated to the pixels of those suspicious regions whose spatial relations with their neighborhood were found inconsistent w.r.t OntoCity’s constraints.

These three channels of data are added to the RGB channels from the next round of learning.

4.3. Classification accuracy results

Two separate classifiers were trained on the training data for each of the two cities used in this work. Each classifier is then applied to the test data for both cities. The classifiers were first trained using a depth concatenation of the RGB channels and three channels that represent the estimations for elevation, shadow, and suspicious areas from the reasoner respectively. The additional channels are set to 0 for the first round of training of the classifiers. On the subsequent iterations, the classifiers are then re-trained with the feedback from the reasoner in the form of adding information to the three additional channels. The process is repeated until the validation accuracy has converged.

The per-class and overall classification accuracy on the test sets for both classifiers before and after the classifiers have been re-trained with the additional information from the reasoner can be seen in Figure 5. The overall accuracy is increased for all combinations of classifier and test data and almost all classes individually. The accuracy on the test data is higher if the classifier was trained on the same city and is decreased if the classifier was trained on another city. The class with the lowest accuracy when the classifier was trained on another city is railroad. The reason for this can be seen by observing that the railroads have different structure and surroundings between the two cities. The reasoner improved the results significantly for the test data on Boden with a classifier trained on the same city, see Figure 5(c).

Some examples of the RGB inputs, predictions, shadow and height estimations for three rounds for Stockholm can be seen in Figure 6. The first round of training of the classifier can give a high number of misclassifications (column 2). When the reasoner has provided shadow estimation (column 5) and elevation information (column 8), the classifier is re-trained and gives an improved classification (column 3). The process is repeated until the validation accuracy has converged and most misclassifications have been corrected (column 4).

The confusion matrix for the last round on both test sets for a classifier that was trained on the Stockholm train data is given in Table 4. The most difficult class to classify is the class railroad and the largest confusion
is between roads and railroad. The semantic referee improved the most for the class road (+20%) at the cost of introducing more confusion between the second largest confusion, which was between vegetation and roads. 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 trees, lawns, parks, grass on the map and hence their elevation values are not informative enough to help the classifier to differentiate them from roads. The accuracy for buildings is already high due to the large amount of buildings in Stockholm and the use of a reasoner did not improve the results for this class. The accuracy for water is also high due to the large amount of water in Stockholm and was improved with the reasoner. One possible reason for this is that the water in the training set contains deeper water that has a darker color compared to the shallow water and channels that are present in Boden.

When the classifier was trained on Boden, which has a smaller amount of training data for buildings and water, but more vegetation than Stockholm, the use of a reasoner improved the accuracy of buildings by 16.9% and water by 20.5% and achieved comparable results with the Stockholm classifier, see Table 5. This illustrates that one of the strengths of using a reasoner is to compensate for classes with a small amount of training data. The reasoner also removes a large amount of previous confusion between buildings and the two classes water and railroad. Similarly, Boden has a large amount of vegetation and therefore already achieves a high accuracy on this class, and while the reasoner improves the results for all classes, it does not further improve the results for vegetation.

The hardware that was used to train the classifiers was a i7-8700K CPU @ 3.70GHz with a GeForce GTX 1070 GPU. The time to train each of the classifiers was around 3 hours.

4.4. Reasoner correction results

We have so far shown how the semantic referee has found a possible correlation between the misclassified regions and their vicinity and reported it back in the form of a semantic augmentation 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 Section 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 water or railroad as the area is externally connected to a building.)

5. Discussion & Future Work

In this work, we proposed 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. Applying spatial and ontological reasoning, the semantic referee is able to semantically augment the misclassifications based on spatial features of regions and the ontological knowledge about cities.

It is worth mentioning that this work relies on what the semantic 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 important to clarify that we do not categorize our current work as a neural-symbolic integrated system, since the neural network algorithm is independent of
Fig. 6. RGB input (column 1), predictions from classifier for three rounds (column 2-4, green=vegetation, gray=road, black=building, blue=water, red=railroad), shadow estimations from reasoner for three rounds (column 5-7, gray=undefined, white=not shadow, black=shadow), and height estimations from reasoner for three rounds (column 8-10, black=low object, white=tall object).

Confusion matrix [%] for both test sets for the classifier that was trained on Stockholm with the use of the reasoner. The numbers in parenthesis show the change from a classifier without using a reasoner.

Table 4

<table>
<thead>
<tr>
<th>Actual label</th>
<th>Predicted label</th>
</tr>
</thead>
<tbody>
<tr>
<td>%</td>
<td>Vegetation</td>
</tr>
<tr>
<td>84.2 (-4.55)</td>
<td>12.9 (+8.26)</td>
</tr>
<tr>
<td>7.4 (-10.9)</td>
<td>86.4 (+20.0)</td>
</tr>
<tr>
<td>1.2 (-0.61)</td>
<td>8.1 (+2.44)</td>
</tr>
<tr>
<td>2.3 (-2.85)</td>
<td>2.4 (-3.41)</td>
</tr>
<tr>
<td>11.0 (-13.8)</td>
<td>37.3 (+15.8)</td>
</tr>
</tbody>
</table>

Confusion matrix [%] for both test sets for the classifier that was trained on Boden with the use of the reasoner. The numbers in parenthesis show the change from a classifier without using a reasoner.

Table 5

<table>
<thead>
<tr>
<th>Actual label</th>
<th>Predicted label</th>
</tr>
</thead>
<tbody>
<tr>
<td>%</td>
<td>Vegetation</td>
</tr>
<tr>
<td>89.2 (-0.16)</td>
<td>7.45 (+0.86)</td>
</tr>
<tr>
<td>11.3 (-4.95)</td>
<td>64.1 (+6.84)</td>
</tr>
<tr>
<td>4.0 (-4.03)</td>
<td>3.62 (-11.9)</td>
</tr>
<tr>
<td>1.25 (-7.53)</td>
<td>4.54 (+3.58)</td>
</tr>
<tr>
<td>10.4 (-8.28)</td>
<td>46.8 (+1.25)</td>
</tr>
</tbody>
</table>

Table 4, Table 5
the symbolic reasoning module, which interacts only with the classifier. This architecture can be viewed as a strength since it allows different types of classifiers to be coupled onto our system in a straightforward manner; thus, our framework remains generic.

For future work, we envision the design of the semantic referee to be tightly integrated in the neural network in such a way that the interaction between the two systems is not limited only to the first and last layers but instead is 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. The work is also supported by Swedish Research Council.

References


