Semantic Web 1 (0) 1–5
IOS Press

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*

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.
Understanding why machine learning algorithms may fail is usually the task of the human expert that uses domain knowledge and contextual information to discover systematic shortcomings in either the data or the algorithm. In this paper, we propose a *semantic referee*, which is able to extract qualitative features of the errors emerging from deep machine learning frameworks and suggest corrections. The semantic referee relies on ontological reasoning about spatial knowledge in order to characterize errors in terms of their spatial relations with in the environment. Using semantics, the reasoner interacts with the learning algorithm as a supervisor. 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, Ontological Reasoning, Spatial Reasoning, 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.

In the context of semantic segmentation and classification for geospatial data, a classifier that uses only the RGB channels as input is error-prone due to the visual similarity between certain classes of data. For example, in satellite imagery, the RGB channels for *water* look similar to *roads* that are covered by a *shadow*, and *buildings* with gray roofs look similar to *roads*. One solution to this problem that has been 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 additional data

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

is not always accessible, e.g., satellite images from Google Maps or several publicly available data sets that only contain the RGB channels. Another possible solution to increase the performance of the classifier is to change the architecture of the network 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, or taking the ad-hoc approach of experimenting with the architecture of the classifier, we propose a method to improve the performance of the machine learning algorithm by focusing on conceptualizing the errors in terms of their spatial relations and surrounding neighborhood. Our method applies 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*, since we use knowledge representation and reasoning methods to arbitrate on the errors arising from the misclassifications.

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[1] standard, and has found its way into various Semantic Web tools and applications over the past few years [10]. 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 [11]. In our work, inspired by the integration of qualitative spatial reasoning methods into imaging procedures as described in [11], we aim to employ spatially-enhanced ontological reasoning techniques in order to assist deep learning methods for image classification via interaction and guidance.

In general, 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] http://www.opengeospatial.org/standards/geosparql

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 interact with deep learning models to improve the classification performance on a city wide scale, as well as a publically available data set.

Our contribution differentiates from the neural-symbolic 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 fist time to be employed for the aforementioned purpose. Finally, our system closes the communication loop between the classifier and the semantic referee.

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 symbolic knowledge represented by ontologies 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, are directly considered and generalized by the learner, whereas in our semantic based model, although the features of the misclassified regions are inferred from the ontology and send back to the classifier, they are not directly applied on the classification output; rather, they are only treated as a new set of data that is sent through the learning process.

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. The classifier is a deep convolutional network with an encoder-decoder structure that outputs an image with the same size as the input image that describes the per-pixel classification. In order to deal with the misclassifications from the classifier, a semantic referee reasons about the errors and consequently provides additional information to the classifier to help it learn from its mistakes and prevent the classifier from making them again. The additional information provided by the reasoner is represented as image channels with the same size as the RGB input and is then concatenated together with the original RGB channels in the depth dimension. In this work, we use three additional channels from the reasoner: the presence of shadows, elevation estimation, and other inconsistencies.

The process of reasoning about the errors includes the conceptualization of the misclassified regions based on their physical (e.g., geometrical) properties.

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

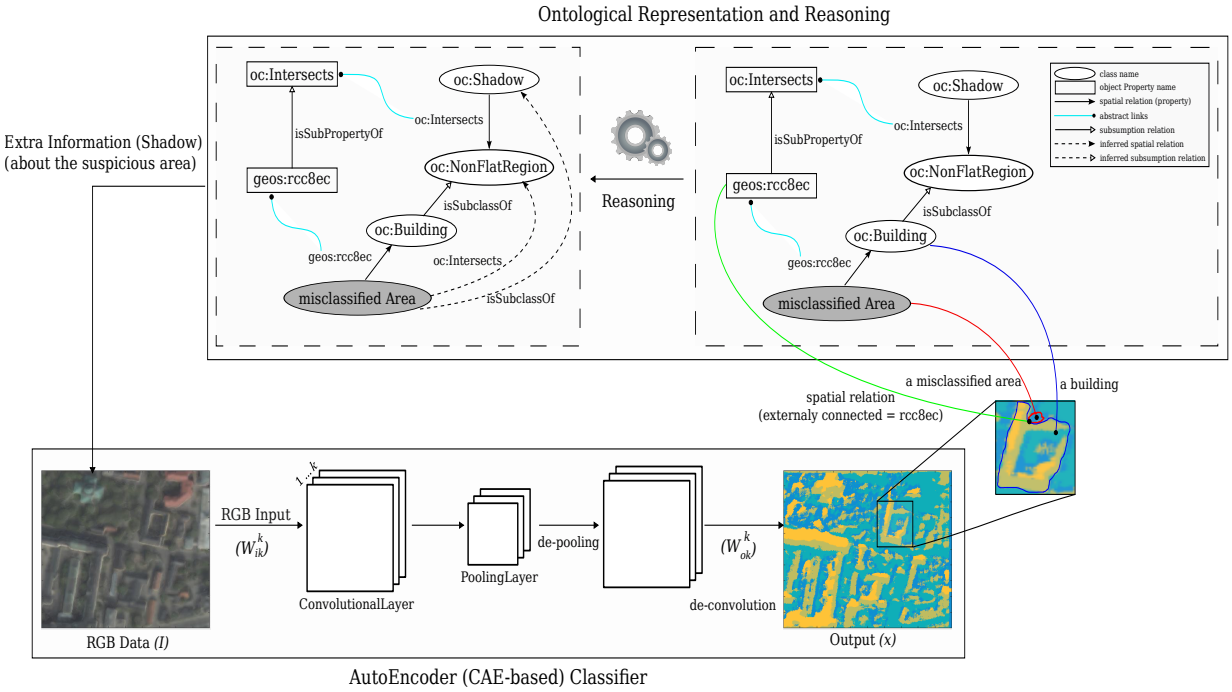


Fig. 1. Overview of applying a semantic referee (top layer) in the form of reasoning upon ontological knowledge to improve the image segmentation task of the classifier (bottom layer). The classifier consists of a deep convolutional network, which receives the RGB data to perform the semantic segmentation. The semantic referee reasons about the mistakes made by the classifier based on ontological concepts and provides additional information back to the classifier that helps prevent the classifier from making the same misclassifications. The ontological knowledge and reasoning methods play the role of the semantic referee that makes sense of the errors from the classifier. For example, the *misclassified region* (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 region, which is the class `oc:shadow` whose constraints are more general ($\exists$ `oc:intersects.oc:NonFlatRegion`) than the spatial representation of the red misclassified region.

The conceptualization process is performed by an ontological reasoner. The reasoner first extracts the geometrical properties of the given misclassified region (e.g., *a building and a road are connected to the misclassified region*) 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 region is then given to the classifier as a referee providing information to be used within the learning process.

The process is then continued by training the classifier on the original RGB data and the new three additional channels from the reasoner, and then the classified regions are sent back to the reasoner to improve the reasoner feedback. The process is repeated until the classification accuracy on the validation data converges. During testing, the same procedure is performed using the same number of iterations that was used during training.

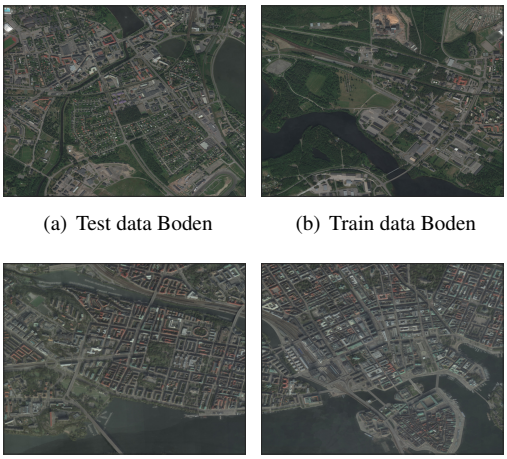
3.2. Data

3.2.1. Stockholm and Boden

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 4000×8000 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.[3]

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 im-

[3]https://www.lantmateriet.se/



(a) Test data Boden (b) Train data Boden

(c) Test data Stockholm (d) Train data Stockholm

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.

balance in the data set, the loss function uses median frequency class weighting.

3.2.2. UC Merced Land Use Dataset

For further evaluation, we use the publicly available UC Merced Land Use dataset [28], which consists of 21 land use classes with 100 images with size 256×256 pixels for each class and a pixel resolution of 1 foot. This data set is labeled by the land use for each image. The work by [29] have labeled each pixel in each image into 17 new classes. The new data set, called DLRSD, that is densely labeled can be used for remote sensing image retrieval (RSIR) and semantic segmentation. For this work, the 17 classes were merged into 8 new classes that were semantically similar in order to reduce the number of classes and resemble the already defined class-rules in the reasoner. The new merged classes, what classes they were merged from, and the prevalence of each new class are seen in Table 2. The introduction of new classes might require some modification in the reasoner. In this work, we have added a rule about the size of the object in the reasoner since the three new classes *airplane*, *car*, and *ship* are generally smaller than the other classes. We removed all images that have *field* or *tennis court* since they require a more complex class definition.

3.3. Data classification

A variation of a Convolutional Auto-encoder (CAE) [30] 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 [31] 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 Lth 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 [32] and has a filter size of 3×3 and 2×2, respectively. The classification of each pixel is performed with a convolution with 5 filters with filter size 1×1 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 [33] and trained using the Adam optimization method [34] 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 an ontological reasoning process. The ontology that we have used as the knowledge model is called OntoCity[4] and contains the domain knowledge about generic spatial constraints in outdoor environments. OntoCity whose (part of) representational details can also be found in [35] 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 [36] 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

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

%	Vegetation	Road	Building	Water	Railroad
Stockholm (train)	7.6	31.3	35.4	23.5	2.2
Stockholm (test)	18.2	36.9	19.7	22.4	2.8
Boden (train)	63.0	19.7	4.9	11.0	1.3
Boden (test)	54.5	25.6	10.8	7.2	1.8

Table 1

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.

New class	Old classes	Prevalence [%]
Vegetation	trees, ~~field~~, grass	28.6
Non-vegetation ground	bare soil, sand, chaparral, ~~court~~	17.6
Pavement	pavement, dock	27.4
Building	building, mobile home, tank	13.7
Water	water, sea	7.9
Airplane	airplane	0.4
Car	cars	2.9
Ship	ship	1.6

Table 2

New merged classes and old classes from the DLRSD data set and class distribution on the new class.

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.

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) [37]:

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

Spatial relations in OntoCity 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:

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

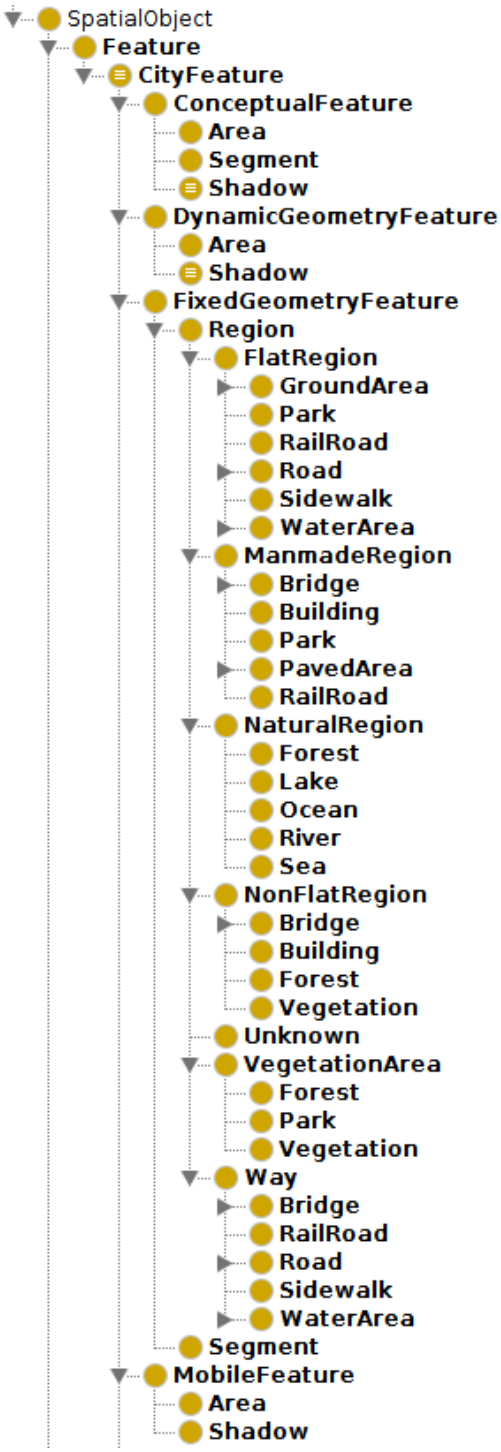


Fig. 3. A snapshot of the hierarchy of concepts in OntoCity. The city features are defined as the subclasses of the `geos:Feature` class defined in GeoSPARQL.

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 per se be categorized into various types such as flat or non-flat, or likewise, into man-made or natural regions:

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

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 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 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:

$$\begin{aligned}
&\texttt{oc:River} \sqsubseteq \texttt{oc:WaterArea} \sqsubseteq \texttt{oc:Region} \\
&\texttt{oc:Road} \sqsubseteq \texttt{oc:PavedArea} \sqsubseteq \texttt{oc:ManmadeRegion} \\
&\texttt{oc:Park} \sqsubseteq \texttt{oc:VegetationArea} \sqsubseteq \texttt{oc:Region} \\
&\texttt{oc:Building} \sqsubseteq \texttt{oc:ManmadeRegion} \sqcap \\
&\quad \texttt{oc:NonFlatRegion}
\end{aligned}$$

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:

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

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):

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

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 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 directly connected to rail roads (referring to the negation of externally connected relation in RCC8: `geos:rcc8ec`).
5. Buildings are not contained by roads (referring to the negation of tangential proper part relation in RCC-8: `geos:rcc8tpp`).
6. Buildings do not contain roads (referring to the negation of tangential proper part inverse relation in RCC-8: `geos:rcc8tppi`).
7. 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. Semantic Augmentation of Errors

The size of each patch of data (either testing or training data for Boden or Stockholm) is is 4000×4000 pixels. We divide each patch into several rectangular segments each is 200×200 pixels. This segmentation is applied to reduce the computational complexity of the reasoning process by only considering the spatial relations of each region with its vicinity i.e., with regions located in a same segment.

The output of the classifier is in the form of labeled pixels. Given a set of pixels carrying same class label, the semantic referee, within a geometrical process[5],

[5]The extraction process is done using the predefined matlab function, *bwboundaries*, used to trace region boundaries in binary images.

first extracts the boundary of these pixels to form a polygon with the region type equivalent to the class label. Each polygon (region) is also assigned with a classification certainty probability. The regions with low certainty are suspected to be misclassified ones and should be prioritized for inspection by the reasoner.

To get the discrete classification of each pixel an argmax is following the softmax layer (that outputs class probabilities). To calculate the predicted class for each region, the class that has the highest mode after the argmax of the class probabilities of each pixel in the region is selected. Then to calculate the classification certainty of the region, we take the average class probabilities of the predicted region class for each pixel in the region.

Given the output of the classification together with ontological knowledge about city features, the reasoner as a semantic referee semantically augments 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: $W = empty, S, m, R$
1: % W: A hash-map, empty in the beginning
2: % S: The given list of rectangular segments
3: % P: The given list of misclassified regions
4: % R: The given list of classified regions
5: **for each** $s \in \mathcal{S}$ **do**
6: $P_s \leftarrow \mathrm{getRegionsInSegment}(P, s)$
7: $R_s \leftarrow \mathrm{getRegionsInSegment}(R, s)$
8: **for each** $r \in \mathcal{R}_s$ **do**
9: $t \leftarrow \mathrm{getRegionType}(r)$
10: **for each** $p \in \mathcal{P}_s$ **do**
11: $q \leftarrow \mathrm{calculateRCC}(p, r)$
12: $W.\mathrm{add}(< q, t >)$
13: **end for**
14: **end for**
15: **end for**
16: $< Q, T > \leftarrow \mathrm{getHighFrequentSpatialRelations}(W)$
17: $C \leftarrow \mathrm{queryOntology}(Q, T)$
18: *augmentation* $\leftarrow \mathrm{getSemantics}(C)$

The algorithm accepts as input the list of segments (S) and the list of both classified (R) and misclassified (P) regions in the form of polygons. 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 region 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 regions for each specific region type (see lines 5-15). In other words, W is defined to contain the geometrical characteristics of the misclassified regions.

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 regions and a specific region type T, then this pair can be generalized and counted as a representative feature of the misclassified regions. 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 ($C \sqsubseteq$ `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 reasoner to localize the shadows.

Furthermore, the ontological reasoner finds many uncertain 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 a three-valued matrix of pixels with the same size as the input data and dedicated to a specific label or landmark on the map defined in the ontology. The elements of each data-channel represent pixels on the map whose values can be either 0, 1 or -1, where: 0 means that the pixel does not belong to any region with the label assigned to the data channel, 1 means the pixel belongs to a region with the label assigned to the data channel, and -1 the label of the pixel is uncertain.

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 regions, as explained in Section 3.5, the reasoner is able to conceptualize the errors. The conceptualization process is based on extracting the spatial relations of the misclassified regions with their segmented neighborhood. This step has been implemented using the open-source JTS Topology Suite[6], whose summary of results for Stockholm and Boden are shown in Table 3 and Table 4. Each cell of the table represents number of misclassified regions 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 regions, considers the pair $< Q, T >$ as the most observed spatial relation Q between the misclassified regions and a specific region type T. Table 3 shows the results of the first round of the classification. As we can see, the most observed spatial relations that involves 136 misclassified regions is the pair $< Q=$ `geos:rcc8ec`, $T=$ `oc:Building`$>$.

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

Table 3

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 `ec` and `po` refer to the RCC-8 relations *externally connected* and *partially overlapping*, respectively.

Type (t) \ Relation (q)	`ec`	`nttpi`
`oc:Building`	**164**	**118**
`oc:RailRoad`	**93**	0

Table 4

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 `ec` and `nttpi` refer to the RCC-8 relations *externally connected* and *non-tangential proper part inverse* (i.e., ogc:contains), respectively.

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 [38]. 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*

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

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 illustrates two samples taken from Stockholm test set classification output where the misclassified regions 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 regions (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 3, between the misclassified regions and the region type `oc: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.

Unlike Stockholm, in classification of Boden test data, most of the extracted spatial relations between misclassified regions and their vicinity were found inconsistent w.r.t. the constraints defined in OntoCity (see Section 3.4.1). As shown in Table 4, 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

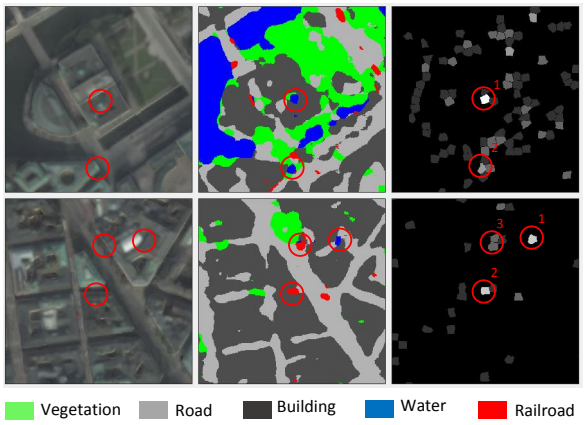


Fig. 4. Two examples of the Stockholm test set classification output along with their input RGB image, classified segmentation and the misclassification. The misclassified regions marked with numbers are in spatial relations with buildings, roads, vegetation, etc. The ontological reasoner can augment the misclassification with the label shadow.

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 uncertain 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 [39]. 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 region, the reasoner is able to localize the shadows as the group of pixels of the misclassified region with the lowest elevation value with respect to the elevation values of the regions intersecting with the misclassified region.

Finally, the third channel of data is dedicated to the pixels of those suspicions 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.

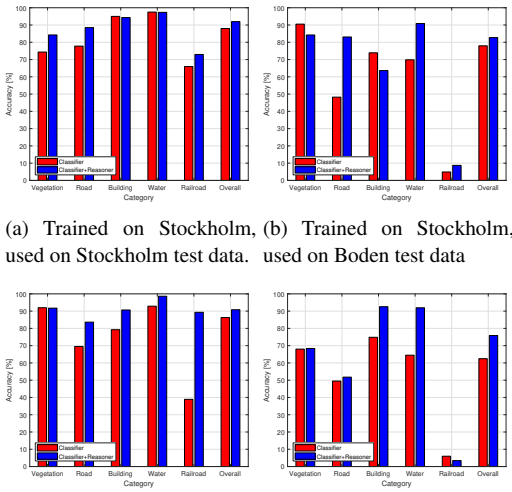
4.3. Classification accuracy results

The following section presents the classification results on Stockholm, Boden, and UC Merced Land Use. The hardware that was used to train the classifiers for was a i7-8700K CPU @ 3.70Ghz with a GeForce GTX 1070 GPU. The time to train each classifier was around 3 hours for all data sets.

4.3.1. Stockholm and Boden

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 uncertain 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 and 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).



(a) Trained on Stockholm, used on Stockholm test data.

(b) Trained on Stockholm, used on Boden test data

(c) Trained on Boden, used on Boden test data

(d) Trained on Boden, used on Stockholm test data

Fig. 5. Classification results on the test data for both cities with two classifiers. The classifiers are trained separately on the training data for both cities before and after using the reasoner.

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 results in 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 of the 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 5. 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

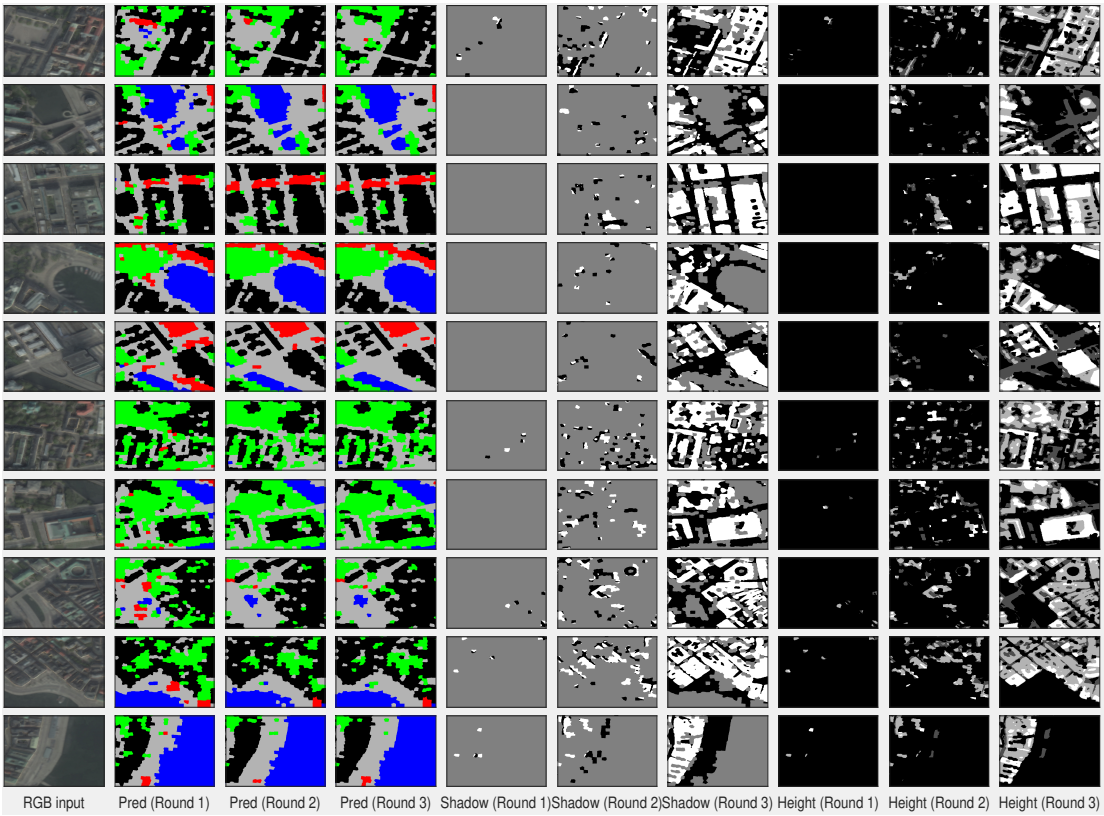


Fig. 6. RGB input (column 1), predictions from classifier for three rounds (column 2-4, green=vegetation, gray=road, black=building, blue=water, red=railroad), shadow estimations from reasoner for three rounds (column 5-7, gray=undefined, white=not shadow, black=shadow), and height estimations from reasoner for three round (column 8-10, black=low object, white=tall object).

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 6. 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.

A summary of the overall classification accuracy can be seen in Table 7. Both classifiers that were trained on Stockholm and Boden, respectively, increase their overall classification accuracy when the additional information from the reasoner is used.

4.3.2. UC Merced Land Use dataset

A new classifier is trained on the UC Merced Land Use dataset. The data was randomly split into 80%/10%/10% training, validation, and test set, respectively. The same structure of the deep convolutional network and training process as the Stockholm and Boden maps were used. The data and the changes to the reasoner is more described in Section 3.2.2. The

		Predicted label			
%	Vegetation	Road	Building	Water	Railroad
Actual label	**84.2 (+4.55)**	12.9 (-8.26)	2.3 (+4.13)	0.39 (-0.28)	0.22 (-0.14)
	7.4 (+10.9)	**86.4 (-20.0)**	5.8 (+8.23)	0.31 (+0.30)	0.11 (-0.63)
	1.2 (+0.61)	8.1 (-2.44)	**90.6 (+1.86)**	0.12 (-0.08)	0.03 (-0.05)
	2.3 (+2.85)	2.4 (+3.41)	0.01 (+0.31)	**95.3 (-6.59)**	0.00 (-0.01)
	11.0 (+13.8)	37.3 (-15.8)	2.6 (+8.0)	0.27 (-0.17)	**48.8 (-5.8)**

Table 5

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 how the result would change compared to a classifier that did not use a reasoner.

		Predicted label			
%	Vegetation	Road	Building	Water	Railroad
Actual label	**89.2 (+0.16)**	7.45 (-0.86)	2.70 (+0.61)	0.39 (-0.10)	0.24 (+0.19)
	11.3 (+4.95)	**64.1 (-6.84)**	24.2 (+1.73)	0.31 (-0.20)	0.07 (+0.36)
	4.0 (+4.03)	3.62 (+11.9)	**92.3 (-16.9)**	0.00 (+0.17)	0.00 (+0.85)
	1.25 (+7.53)	4.54 (-3.58)	0.14 (+16.6)	**94.1 (-20.5)**	0.00 (+0.01)
	10.4 (+8.28)	46.8 (-1.25)	6.92 (+10.2)	0.04 (+0.18)	**35.8 (-17.4)**

Table 6

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 how the result would change compared to a classifier that did not use a reasoner.

Training data	Overall accuracy [%] with reasoner	Overall accuracy [%] without reasoner
Stockholm	86.7	81.4
Boden	81.3	73.2

Table 7

Overall classification accuracy for the both classifier trained on only Stockholm or Boden training data for the test data from both cities with and without the use of a reasoner.

training went through three rounds of training between the classifier and the reasoner. The per-class and overall classification accuracy on the test set of the first and final round can be seen in Table 8. All classes get an improved classification accuracy except *non-vegetaion ground*. The largest improvement is for the classes *airplane* and *car*.

The input image, predictions, shadow estimation, and height estimation for 6 test images from the UC Merced Land Use dataset can be seen in Figure. 7. The second column shows the predictions without using a reasoner and the third column shows the predictions when using the reasoner after three iterations of training. The predictions have been averaged within each region to reduce noise. The forth and fifth columns show the shadow and height estimations from the reasoner. From the first four images it can be seen that the reasoner helps to more accurately predict *cars* (a class that should have a small size). It can also be seen that many misclassifications that were predicted as *airplane* have been changed with the use of a reasoner (a class that should have a larger size). There is still some confusion between *pavement* and *non-vegetation ground* that seems to not have been captured semantically in the reasoner. From the last image we see that the reasoner removes some of the confusion between gray roofs from a *building* and *pavement*.

Class	Accuracy [%] without reasoner	Accuracy [%] with reasoner
Vegetation	76.62	81.01
Non-vegetation ground	81.74	75.18
Pavement	78.43	78.96
Building	81.32	84.52
Water	80.64	87.08
Airplane	68.10	76.34
Car	62.71	94.64
Ship	90.64	98.86
Mean accuracy	77.53	83.29

Table 8

Per-class and mean classification accuracy on the testing set on UC Merced Land Use dataset with and without the use of a reasoner.

4.4. Reasoning for improved learning

So far, the reasoner does not correct misclassifications directly but rather directs the classifier to improve its learning about categories. The reason for this approach is twofold. First, our objective is to ultimately learn about spatial features about regions, and the rea-

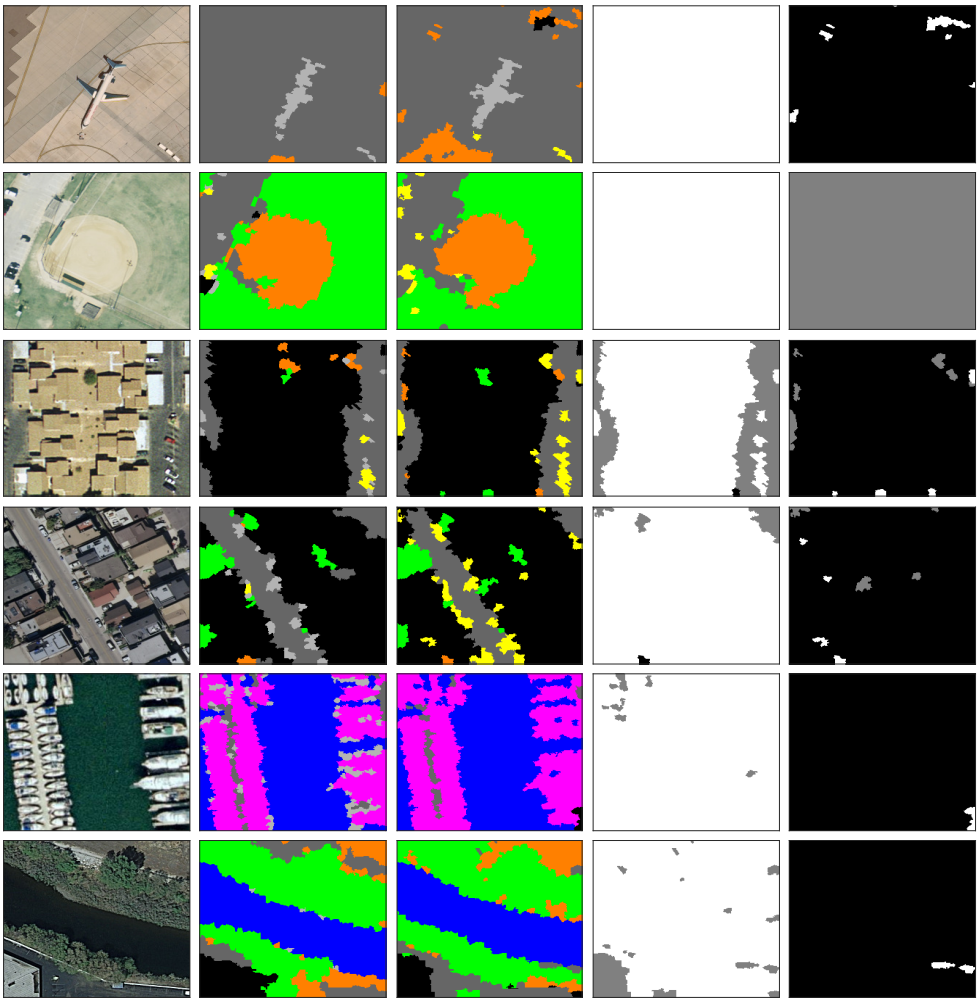


Fig. 7. Input image (column 1), predictions from classifier without and with using a reasoner (column 2-3, green=vegetation, orange=non-vegetation ground, gray=pavement, black=building, blue=water, yellow=car, purple=ship, light gray=airplane), shadow estimation from reasoner (column 4, gray=undefined, white=not shadow, black=shadow), and height estimation from reasoner (column 5, black=low height, white=large height).

soner is used to automate the role of a supervisor. Secondly, the reasoner as a referee is also inherently uncertain, and thus may provide several candidate labels for a region. Therefore, the integration between the reasoning and learning architecture has been done in manner described in this paper.

4.5. Evaluation of elevation estimation from the reasoner

For data sets that already contain some of the information that the reasoner can provide, it is justified to add it directly as input to the classifier instead of estimating it with the reasoner. One such feature is height information that could come from a Digital Surface Model (DSM), which was available for the maps of Stockholm and Boden, but not for the UC Merced Land Use data set. Table 9 shows a comparison of the classification accuracy between a classifier that was trained on RGB and elevation information from a DSM and a classifier that was trained on RGB and elevation information from the reasoner. It can be seen that the classifier that was trained on the groundtruth DSM gave slightly better overall classification accuracy. However, the reasoner-provided elevation gives comparable results and is even better for some classes (*vegetation* and *building*). This shows that using a reasoner is a viable replacement for data sets that do not

contain elevation information. Furthermore, a reasoner can provide other features, such as shadow.

Class	Accuracy [%] with reasoner elevation	Accuracy [%] with DSM elevation
Vegetation	74.16	69.78
Road	68.37	75.76
Building	93.33	89.80
Water	91.43	95.11
Railroad	42.12	54.88
Mean accuracy	80.64	82.79

Table 9

Per-class and mean classification accuracy on the testing set on Stockholm when trained on Stockholm RGB and elevation from reasoner or with groundtruth DSM.

5. Discussion & Future Work

In this work[7], we proposed an ontological 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 geometrical processing 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 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.

[7]The source code can be found at: https://github.com/marycore/SemanticRobot

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

[1] X.X. Zhu, D. Tuia, L. Mou, G. Xia, L. Zhang, F. Xu and F. Fraundorfer, Deep learning in remote sensing: a review, *CoRR* **abs/1710.03959** (2017). http://arxiv.org/abs/1710.03959.

[2] K. Janowicz, S. Scheider and B. Adams, A Geo-semantics Flyby, in: *Reasoning Web*, Lecture Notes in Computer Science, Vol. 8067, Springer, 2013, pp. 230–250.

[3] 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).

[4] 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.

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

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

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

[17] 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.

[18] M. Alirezaie, M. Längkvist, M. Sioutis and A. Loutfi, A Symbolic Approach for Explaining Errors in Image Classification Tasks, in: *IJCAI-ECAI-2018 Workshop on Learning and Reasoning Principles & Applications to Everyday Spatial and Temporal Knowledge*, 2018.

[19] 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.

[20] 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.

[21] 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.

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

[23] 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.

[24] 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.

[25] R. Liu, J. Lehman, P. Molino, F. Petroski Such, E. Frank, A. Sergeev and J. Yosinski, An Intriguing Failing of Convolutional Neural Networks and the CoordConv Solution, *ArXiv e-prints* (2018).

[26] Z. Wang and O. Veksler, Location Augmentation for CNN, *arXiv preprint arXiv:1807.07044* (2018).

[27] S. Minton, J.G. Carbonell, C.A. Knoblock, D.R. Kuokka, O. Etzioni and Y. Gil, Explanation-based learning:A problem solving perspective, *Artificial Intelligence* **40** (1989), 63–118.

[28] Y. Yang and S. Newsam, Bag-of-visual-words and spatial extensions for land-use classification, in: *Proceedings of the 18th SIGSPATIAL international conference on advances in geographic information systems*, ACM, 2010, pp. 270–279.

[29] Z. Shao, K. Yang and W. Zhou, Performance Evaluation of Single-Label and Multi-Label Remote Sensing Image Retrieval Using a Dense Labeling Dataset, *Remote Sensing* **10**(6) (2018), 964.

[30] 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.

[31] O. Ronneberger, P. Fischer and T. Brox, U-Net: Convolutional Networks for Biomedical Image Segmentation, in: *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, N. Navab, J. Hornegger, W.M. Wells and A.F. Frangi, eds, Springer International Publishing, Cham, 2015, pp. 234–241. ISBN 978-3-319-24574-4.

[32] 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.

[33] 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.

[34] D.P. Kingma and J. Ba, Adam: A Method for Stochastic Optimization, *CoRR* **abs/1412.6980** (2014).

[35] 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.

[36] 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.

[37] 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.

[38] 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.

[39] 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.