EXPLORING SEMANTIC RELATIONSHIPS FOR HIERARCHICAL LAND USE CLASSIFICATION BASED ON CONVOLUTIONAL NEURAL NETWORKS

C. Yang *, F. Rottensteiner, C. Heipke

Institute of Photogrammetry and GeoInformation, Leibniz Universität Hannover - Germany
{yang, rottensteiner, heipke}@ipi.uni-hannover.de

Commission II, WG II/6

KEY WORDS: hierarchical land use classification, CNN, geospatial database, aerial imagery, semantic relationships

ABSTRACT:

Land use (LU) is an important information source commonly stored in geospatial databases. Most current work on automatic LU classification for updating topographic databases considers only one category level (e.g. *residential* or *agricultural*) consisting of a small number of classes. However, LU databases frequently contain very detailed information, using a hierarchical object catalogue where the number of categories differs depending on the hierarchy level. This paper presents a method for the classification of LU on the basis of aerial images that differentiates a fine-grained class structure, exploiting the hierarchical relationship between categories at different levels of the class catalogue. Starting from a convolutional neural network (CNN) for classifying the categories of all levels, we propose a strategy to simultaneously learn the semantic dependencies between different category levels explicitly. The input to the CNN consists of aerial images and derived data as well as land cover information derived from semantic segmentation. Its output is the class scores at three different semantic levels, based on which predictions that are consistent with the class hierarchy are made. We evaluate our method using two test sites and show how the classification accuracy depends on the semantic category level. While at the coarsest level, an overall accuracy in the order of 90% can be achieved, at the finest level, this accuracy is reduced to around 65%. Our experiments also show which classes are particularly hard to differentiate.

1. INTRODUCTION

Land use (*LU*) describes the socio-economic function of a piece of land. This information is usually collected in geospatial databases, often acquired and maintained by national mapping agencies. The objects stored in these databases are typically represented by *polygons* with categories indicating the object's LU. To keep such databases up-to-date, the content can be compared with new remote sensing data. If the new data contradict the database content for a specific object, the object class label in the database needs to be updated. To automate this process, a class label related to its LU has to be determined from the remote sensing data for every object in the database. Typically, this is achieved in a procedure consisting of two steps: first, the imagery is used to predict the land cover for each pixel; the land cover results and the images are combined in a second classification process to determine the LU for every database object (Gerke et al., 2008; Helmholtz et al., 2012). In this context, supervised classification methods are frequently applied, most recently based on Convolutional Neural Networks (CNN) (Zhang et al., 2018; Yang et al., 2019), which have been shown to outperform other classifiers such as Conditional Random Fields (CRF) (Albert et al., 2017).

One problem of existing methods for LU classification is that they only differentiate a small number of classes, while the object catalogues of LU databases may be much more detailed. For instance, in the LU layer of the German cadastre, about 190 categories are differentiated (AdV, 2008). Clearly, this catalogue contains object types that cannot be expected to be differentiated from remote sensing data, but of course, the usefulness of an automatic approach grows with an increasing number of class labels. It is an important fact that many topographic databases contain LU information in different semantic levels of abstraction. At the coarsest level, only a few broad classes such as *settlement*, *traffic* or *vegetation* are differentiated. At the finer levels, these classes are hierarchically refined, and the full number of different categories is only differentiated at the finest level of the class structure. Fig. 1 shows two examples for database objects with corresponding imagery and the annotations from the first three levels of the object catalogue in (AdV, 2008).

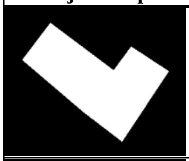
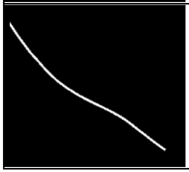
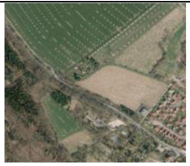
Object shape	RGB image	Size	L	Category
		320 x 260 pixels	I	*residential*
			II	*recreation area*
			III	*graveyard*
		3400 x 3100 pixels	I	*traffic*
			II	*road traffic*
			III	*motor road*

Figure 1: Two database objects with images (rescaled) and categories in three semantic layers. *L:* semantic layer starting from the coarsest (*I*) to the finest (*III*).

Albert et al. (2016) investigated the maximum level of semantic resolution that their CRF-based LU classification could achieve. They divided the land use categories into two levels, both corresponding to mixtures of the three coarsest semantic levels according to (AdV, 2008). Starting from a classification of the coarse level, they refine one coarse category after the other: in a greedy iterative procedure one category is split into the maximum set of sub-categories and then sub-categories are merged if the results indicate they cannot be separated. As a result, Albert et al.

* Corresponding author.

(2016) obtain a class structure consisting of a mixture of 10 categories from different semantic levels of the object catalogue, and conclude that this is the largest set of classes that can be separated using their approach. In this paper we take a different direction. We propose to predict the LU categories of multiple semantic levels *simultaneously* using a CNN-based approach. In this context, we exploit the intrinsic relations between the categories at different layers, which leads to *hierarchical LU classification*. In our method, the hierarchical relations are explicitly integrated into the CNN for training and inference. To achieve our goals, we expand the existing two-step procedure of (Yang et al., 2019) to this hierarchical setting, adapting a method proposed by Hu et al. (2016) for learning structured inference neural networks of natural images by modelling label relationships for our purposes. The input consists of high-resolution aerial imagery, a land cover layer obtained by semantic classification and derived data such as a Digital Surface Model (DSM) and a Digital Terrain Model (DTM). The scientific contribution of this paper can be summarized as follows:

- We expand a CNN-based method for the classification of LU to predict LU categories at multiple semantic levels simultaneously, sharing the feature extraction part of the network and adding independent classification heads; this corresponds to a multi-task learning approach, e.g. (Leiva-Murillo et al., 2013). Furthermore, inspired by (Hu et al., 2016), we propose to improve this multi-task method by additional connections between the semantic layers so that the new method incorporates the semantic relations between the different hierarchical levels.
- Based on the multi-task learning network, we propose two additional network variants to guarantee hierarchically consistent predictions. One variant starts from the predictions of the coarsest level and adapts the predictions in the finer levels to be consistent, and the other one works in the opposite direction. For training the two variants, two novel objective functions are proposed.
- We conduct an extensive set of experiments to compare these network variants, to highlight the benefits of considering the relations between the different semantic levels and to investigate the limits of the proposed approaches in differentiating finer class structures.

In section 2, we give a review of related work. Our approach for hierarchical land use classification is presented in section 3. Section 4 describes the experimental evaluation of our approach. Conclusions and an outlook are given in section 5.

2. RELATED WORK

We start this review with an overview of LU classification techniques before discussing hierarchical classification methods.

As pointed out earlier, methods for LU classification usually apply a two-step procedure: first, the land cover is determined based on the given image data, and then the land cover together with image and derived data (e.g. a DSM) serves as input for LU classification. Traditionally, hand-crafted features are derived from input data. These features may quantify the spatial configuration of the land cover elements within a land use object, describing the size and shape of the land cover segments (Hermosilla et al., 2012). Other features are based on the frequency of local spatial arrangements of land cover elements within a land use object (Novack and Stilla, 2015), applying the adjacency-event matrix (Barnsley & Barr, 1996; Walde et al., 2014). Supervised classifiers applied in this context include Support Vector Machines (Montanges et al., 2015) and Random Forests (Albert et al., 2017), the latter also embedded in contextual models like Conditional Random Fields (CRF).

Since the success of AlexNet (Krizhevsky et al., 2012), CNN, replacing hand-crafted features by a representation learned from training data, have been shown to outperform other classifiers. They have also been adopted in remote sensing (Zhu et al., 2017). In this context, a big challenge for applying CNN for the prediction of class labels for LU polygons is the large variation of polygon shapes and sizes. To the best of our knowledge the first work classifying LU objects from a geospatial database by CNN is (Yang et al., 2018). The authors decompose large polygons into multiple patches that can be classified by a CNN. However, they extract the employed image and land cover data inside the polygon and set the areas outside to 0, which leads to a loss of context information. Yang et al. (2019) extend this approach by constructing a representation of a polygon by a binary mask while using image data for the entire window to be classified. In this paper, we adapt their basic framework, but extend the LU classification by considering class labels at different semantic levels. Zhang et al., (2018) proposed a method to classify urban land use objects by applying two CNNs. They perform image segmentation and then use the segmentation results to obtain polygons based on which the inputs for the two CNNs are generated. However, they focus only on urban scenes, without any consideration on rural areas. Zhang et al., (2019) propose a joint deep learning framework for land cover and land use classification where they use multi-layer perceptions for land cover classification and a CNN for land use classification based on Zhang et al. (2018). They differentiate a set of about 10 LU classes in their experiments without further investigations concerning the semantic resolution that can be achieved.

Albert et al. (2016) propose a method based on CRF to investigate the maximum level of semantic resolution that can be achieved, applying the greedy refinement strategy outlined earlier, but their goal is to define a suitable class structure rather than using the hierarchical structure of the object catalogue in a systematic way. Considering multiple semantic levels of categories can result in the prediction of multiple labels per object, which can pose a problem. This issue is tackled in (Hua et al., 2019). The authors propose a method for multi-label classification of aerial images by applying a CNN with LSTM (Long Short Term Memory) cells. The goal is to predict a set of labels for one input image, describing each object type that appears in that image. No semantic relations between the labels are modelled explicitly. Therefore, the method cannot be directly transferred to our problem. Different semantic levels of categories can also be dealt with as different categories, and the intrinsic relation of the different levels could be tackled by multi-task learning approaches, e.g. (Leiva-Murillo et al., 2013), though this seems not to have been done yet. In computer vision, many approaches dedicated to the classification of images with semantic relations between categories exist. Deng et al. (2014) propose the first CNN-based work for classification with semantic relations between different class labels. They define a HEX (Hierarchy and Exclusion) graph to model different types of semantic relations: two labels may have a hierarchical relation; they may be exclusive or overlapping. The CNN only has one output layer for all classes, but the HEX graph is considered in both, training and inference to achieve a consistent classification result, e.g. to ensure that an image cannot be classified as showing a cat and a specific dog breed at the same time. However, this results in a very complex training and inference procedure. Guo et al. (2018) propose a CNN-RNN (recurrent neural network) strategy to address hierarchical classification. A

CNN acts as a feature extractor and is trained to predict class labels at the coarse semantic level. Then, the CNN features and the output of the coarse level are fed into a RNN structure which is used to propagate the information from the coarse level to finer labels. Nonetheless, information is only predicted from the coarse level to the finer labels. Hu et al. (2016) propose a network based on a CNN for hierarchical classification in three levels, using a bidirectional message passing mechanism from the class scores of the coarse category to the class scores of the fine category and vice versa. Thus, the class scores of each level are enhanced considering information from other levels of the hierarchy. However, the message passing is done only between neighbouring levels. Though embedded in a completely different context, the method proposed in this paper is inspired by Hu et al. (2016). However, we argue that for a specific category level, all its ancestor levels and descendant levels are helpful for its identification. Thus, we adapt the message passing, so that the class scores of one level receive messages from all ancestor levels and all descendant levels. More importantly, we can guarantee consistency of the predictions with the class hierarchy.

3. CNN-BASED HIERARCHICAL CLASSIFICATION

The first input required for our method consists of a LU database in which objects are represented by polygons with LU categories at multiple semantic levels according to a hierarchical object catalogue. Furthermore, a multispectral aerial image (R, G, B, NIR), a normalised DSM (nDSM, i.e., the difference between a DSM and a DTM) and pixel-wise class scores for land cover from a previous classification step are required. In order to produce the latter, we use the CNN-based method of Yang et al. (2019), which delivers a vector of class scores for every pixel of the input image (one entry per land cover class). The input polygon is used to generate a binary object mask aligned with the image grid. The goal of the proposed method is to predict one class label per semantic level for each LU object, extending our previous work (Yang et al., 2019). While these labels are known for some of the polygons, which can be used for training the CNN, they are to be determined for the rest.

In CNN-based LU classification, the large variation of polygons in terms of their geometrical extent is a challenge (see examples for a very large road and a small residential object in Fig. 1), because a CNN requires a fixed input size for the image to be classified (256 x 256 pixels in our case). The way in which the image patches of that size are prepared is described in section 3.1. Section 3.2 outlines the basic CNN structure, introducing a multitask learning scenario for LU classification at different semantic levels, while Section 3.3 describes several network variants that hierarchically interact in training and classification.

3.1 Patch preparation

The basic approach to prepare the input data is to extract a window of 256 x 256 pixels centred at the centre of gravity of the object from all data (image and DSM, binary object mask, land cover scores) and present it to the CNN. This is unproblematic if the polygon size corresponds well to the window size at the ground sampling distance (GSD); otherwise the window is either dominated by information outside the object (for very small objects) or the object does not fit into the window. The method we adopt to cope with the latter problem is *tiling*: we split the window enclosing the object into tiles (patches) of the desired size and classify all patches having a meaningful overlap with the object independently. Afterwards, the results for the individual input patches are combined (cf. section 3.3).

3.2 Baseline CNN architecture

The basic network architecture we use for LU classification is based on the *LuNet* architecture (Yang et al., 2019). *LuNet* consists of a series of convolutional and pooling layers before being split into two branches. The first branch consists of a set of convolutional and pooling layers while the second branch (ROI location layer) extracts a region of interest from the feature map, rescales it and applies convolutions and pooling to that rescaled feature map. Before the classification layer, the feature vectors of the two branches are concatenated; for more details, we refer the reader to (Yang et al., 2019). We keep the entire architecture except for the single classification layer, which is replaced by *B* classification layers (one layer per semantic category level). The resulting structure is shown in Fig. 2 for *B* = 3 levels. This structure corresponds to a variant of multi-task classification (Leiva-Murillo et al., 2013): the predictions of the labels at different semantic levels are considered to be different classification tasks; the prediction itself is independent, but based on a shared (512 dimensional) feature vector extracted from the input data. The parameters of all components of the network are determined simultaneously. Thus, the CNN learns to produce a representation that is meaningful for all tasks.

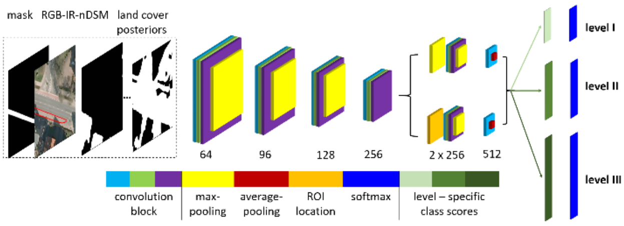


Figure 2. Main architecture of *LuNet-MT* for *B* = 3 semantic levels (level I / coarsest level - level III / finest level).

Integration of the semantic dependencies: Given the object catalogue, the relationships between semantic levels are known. To add this prior knowledge to the network, we propose to expand the network structure so as to consider the semantic dependencies. Starting from Fig. 2, we identify each category level by a roman numeral from the coarsest level I and increasing the number as the semantic resolution is increased. For each semantic level *l*, the classification head consists of one fully connected (FC) layer that delivers a vector of un-normalized class scores $\boldsymbol{z^l} = \left(z_1^l, \dots, z_{M_l}^l\right)^T$, where $C_c^l = \{C_1^l, \dots, C_{M_l}^l\}$ is a set of LU classes at category level *l* and z_c^l is the class score of an image *X* for class C_c^l. Based on the un-normalized class scores $\boldsymbol{z^l}$**,** the expansion of the network structure is shown in Fig. 3. There are two additional layers per semantic layer, each with a specific structure of connections to the previous layer: First, information is passed on from coarser levels to finer levels; after that, information is passed back from finer levels to coarser levels. The expanded network is referred to as *LuNet-MT* (MT for multi-task) in the remainder.

In the first of the two additional layers, we produce a set of intermediate class scores $\boldsymbol{z^l_{mid}}$ at each level *l*, where the class score at each level except the first (coarsest) one receives input from the same or from all coarser levels in the previous layer of the network. For the coarsest level (*l* = 1), the scores from the previous layer are copied, i.e. $\boldsymbol{z^1_{mid}} = \boldsymbol{z^1}$. Otherwise, $\boldsymbol{z^l_{mid}}$ is computed according to:

$$\boldsymbol{z^l_{mid}} = W^l \cdot f(\boldsymbol{z^l}) + \textstyle\sum_{i=1}^{l-1}[f(W_i^{pos,l} \cdot f(\boldsymbol{z^i})) - f(W_i^{neg,l} \cdot f(\boldsymbol{z^i}))] \, , \quad (1)$$

where $f()$ is the ReLU activation function and W^l as well as $W_i^{pos,l}, W_i^{neg,l}$ are the parameters of that layer that are to be learned in training along with the other parameters of the network. Here, the superscripts *pos* and *neg* specify positive and negative semantic relationships. If a category is divided into multiple sub-categories at a finer level, these sub-categories are *positively* related to it; a category is *negatively* related to sub-categories at a finer level if they are not derived from it. In $W_i^{pos,l}, W_i^{neg,l}$, only the parameters with the specific relationships are learned and the others are set to 0.

In the second additional layer, we produce the final un-normalized class scores $\boldsymbol{z}_{out}^{l}$ at each level l. Here, the class score at each level except the last (finest) one receives input from the same or from all finer levels in the previous layer. For the finest level (l = B), the scores from the previous layer are copied, i.e. $\boldsymbol{z}_{out}^{B} = \boldsymbol{z}_{mid}^{B}$. Otherwise, $\boldsymbol{z}_{out}^{l}$ is computed according to:

$$\boldsymbol{z}_{out}^{l} = V^{l} \cdot f(\boldsymbol{z}_{mid}^{l}) + \sum_{j=l+1}^{B}[f(V_j^{pos,l} \cdot f(\boldsymbol{z}_{mid}^{j})) - f(V_j^{neg,l} \cdot f(\boldsymbol{z}_{mid}^{j}))] , \quad (2)$$

where V^{l} and $V_j^{pos,l}, V_j^{neg,l}$ are the parameters of that layer and $f()$ is the ReLU function. The superscripts *pos* and *neg* have the same meaning as in eq. 1. Finally, the un-normalized class scores are passed through a softmax layer to obtain probabilistic scores, i.e., for each layer, $\boldsymbol{z}_{out}^{l}$ is used as the argument of the softmax function.

$$P(C_c^l|\mathrm{X}) = softmax(\boldsymbol{z}_{out}^{l}, C_c^l) = \frac{exp(z_{out,c}^{l})}{\sum_{m=1}^{M_l} exp(z_{out,m}^{l})}, \quad (3)$$

Training is based on stochastic mini-batch gradient descent (SGD) with weight decay and step learning policy; the objective function is the *extended focal loss* (Yang et al., 2019):

$$L = -\frac{1}{N} \cdot \sum_{l,c,k} \left[y_c^{l,k} \cdot (1 - P\left(C_c^l|X_k\right))^{\gamma} \cdot log\left(P\left(C_c^l|X_k\right)\right)\right], \quad (4)$$

where X_k is the k^{th} image in a mini-batch, N is the number of images in a mini-batch, and $y_c^{l,k}$ is 1 if the training label of X_k is C_c^l in level l and 0 otherwise. More details about training are given in Section 4.1.

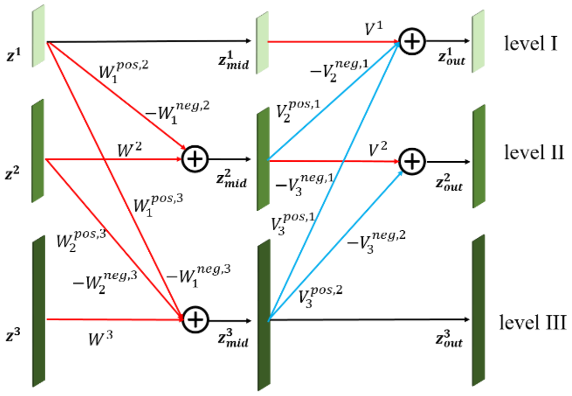


Figure 3. Expanded classification head of *LuNet-MT*. Please refer to the text for the explanation of the variables. The leftmost green bars correspond to the green bars containing the class scores in Fig. 2. Please note that ReLU activation is not shown here.

3.3 Network variants and implementation

LuNet-MT obtains predictions of multiple semantic levels simultaneously while exploring the semantic dependencies explicitly. However, the predictions are not guaranteed to be consistent with the object catalogue hierarchy. For instance, one object predicted as *settlement* at the coarse level could be predicted as *road traffic* at the fine level. Obviously, these two predictions are not hierarchically related. To obtain predictions that are consistent with the class hierarchy, two strategies for *hierarchical training and inference* are proposed. The first one is referred to as *coarse-to-fine* (*C2F*). Using this strategy, we first predict the categories at the coarsest level (I) and use them to control the predictions at the finer levels. During inference, only the un-normalized scores of the sub-categories at a finer level which are derived from the predicted category at the coarser level are used as input of the softmax function to obtain probabilistic scores. During training, the ground truth labels of coarser levels are used to select the un-normalized scores at the finer level. The second strategy is referred to as *fine-to-coarse* (*F2C*). Here, we first predict the categories at the finest level (III). Then we select the category of which the category at the finest level is a sub-class as its prediction at the coarser level. An illustration of the two approaches is shown in Fig. 4. Note that if the first predictions in the C2F approach are wrong, the subsequent predictions at the finer levels will be wrong as well. Nonetheless, in the F2C approach, there is still chance to obtain right predictions at the coarser levels if the first predictions are wrong. Relying on the two approaches, two network variants based on *LuNet-MT* are proposed.

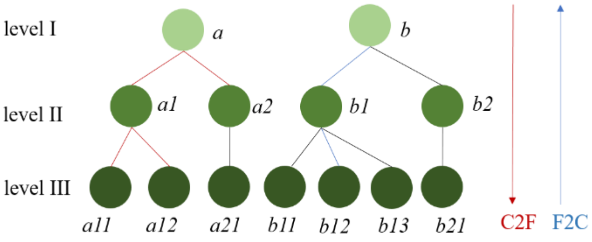


Fig. 4: Illustration of the *C2F* and *F2C* approaches (see main text for a description of the two strategies). The lines between levels indicate hierarchical relations between classes at different semantic levels. *a*, *b* are classes at level I, the classes at the subsequent levels are sub-classes of *a* and *b*, respectively.

3.3.1 *HierLuNet-C2F*: this variant realizes the C2F strategy. The probabilistic scores at the finer levels are:

$$P'(C_c^l|\mathrm{X}) = softmax(\boldsymbol{z}_{out}^{sub,l}, C_c^l) = \frac{exp(z_{out,c}^{sub,l})}{\sum_{m=1}^{M_l} exp(z_{out,m}^{sub,l})}, if\ l > 1, \quad (5)$$

$\boldsymbol{z}_{out}^{sub,l}$ are the un-normalized scores in level l consistent with the coarser level. Together with the class scores $P(C_c^1|\mathrm{X})$ of the coarsest level, these variants of the class scores are plugged into eq. 4 for optimization.

3.3.2 *HierLuNet-F2C*: this variant realizes the F2C strategy. First, the probabilistic scores of the finest level (III) are determined using eq. 3. For the coarser levels (I and II), softmax is not suitable to obtain the probabilistic scores, because the classes have to be the ancestors of the class at level III and, consequently, the predictions are known. Thus, we apply the sigmoid function to the corresponding un-normalized scores to generate normalized scores.

$$\hat{P}(C_c^l|\mathrm{X}) = sigmoid(z_{out}^{c,l}), if\ l < B, \quad (6)$$

During training, the objective function consists of two parts: for the finest level, it is the same as eq. 4, referred to as L_{III}, and for the coarser levels (I and II, $l < B$), the objective function is:

$$L_{I,II} = -\frac{1}{N} \cdot \sum_{l,c,k} \left\{ \begin{array}{c} y_c^{l,k} \cdot \tilde{y}_c^{l,k} \cdot \left(1 - \hat{P}(C_c^l|X_k)\right)^{\gamma} \cdot log\left(\hat{P}(C_c^l|X_k)\right) + \\ (1 - y_c^{l,k} \cdot \tilde{y}_c^{l,k}) \cdot [y_c^{l,k} \cdot \left(1 - \hat{P}(C_c^l|X_k)\right)^{\gamma} \cdot log\left(\hat{P}(C_c^l|X_k)\right) + \\ (1 - y_c^{l,k}) \cdot \left(\hat{P}(C_c^l|X_k)\right)^{\gamma} \cdot log\left(1 - \hat{P}(C_c^l|X_k)\right)] \end{array} \right\}, \quad (7)$$

https://doi.org/10.5194/isprs-annals-V-2-2020-599-2020 | © Authors 2020. CC BY 4.0 License.

where $\tilde{y}_c^{l,k}$ is 1 if the prediction of image X_k.is class C_c^l in level l and 0 otherwise. If the prediction matches the ground truth (i.e. $y_c^{l,k} = \tilde{y}_c^{l,k} = 1$) , the probabilistic score of class C_c^l is to be maximized; otherwise, the probabilistic score of the referenced category is to be maximized and the others are to be minimized. The sum of $L_{III} + L_{I,II}$ is used for optimization.

3.3.3 Inference at object level: The inference of the objects which are not split during tiling is straightforward by using the prediction of the related patches. The inference of objects which had to be split (termed as *compound* objects) differs in the different network variants. In variant *LuNet-MT,* for a compound object, the product of the probabilistic class scores of the patches in each individual semantic level is computed. Subsequently, the product is used for obtaining the predicted label. In variant *HierLuNet-C2F*, for a compound object, the prediction in the coarsest level (I) is made by a majority vote of the predictions of its patches. To guarantee hierarchical consistency, the predictions in the finer levels are sorted in a descending order according to their occurrences. Searching the predictions based on the order is undertaken and the best one which is a sub-category of the prediction in the coarser level is considered as the predicted label. Finally, in variant *HierLuNet-F2C*, for a compound object, the prediction of the finest level (III) is taken by majority vote of the predictions of the related patches. The prediction procedure of the coarser levels is similar to the one in *HierLuNet-C2F*, but in the opposite direction, so that hierarchical consistency is guaranteed.

3.3.4 Implementation: all networks are implemented based on the tensorflow framework (Abadi et al., 2015). We use a GPU (Nvidia TitanX, 12GB) to accelerate training and inference.

4. EXPERIMENTS

4.1 Test Data und Test Setup

4.1.1. Test Data: We use two German test sites for our experiments. The first one is located in Hameln. It covers an area of 2 x 6 km^2 and shows various urban and rural characteristics. The other one is located in Schleswig, covering an area of 6 x 6 km^2 and having similar characteristics as Hameln. For both test sites, digital orthophotos (DOP), a DTM, a DSM derived by image matching and land use objects from the German Authoritative Real Estate Cadastre Information System (ALKIS) are available. The DOP are multispectral images (RGB + infrared / IR) with a GSD of 20 cm. We generated a normalised DSM (nDSM) by subtracting the DTM from DSM. The reference for land use objects was derived from the geospatial database. To obtain the hierarchical class structure, we follow the ALKIS object catalogue (AdV, 2008). The details of the hierarchical class structure along with the number of samples are presented in Tab. 1. Note that the class structures for the two test sites are slightly different because some classes only occur in one test site. In level I, the structures are the same with 4 categories. In level II, although there are 15 categories, both test sites only contain samples for 14 categories: in Schleswig, there is no sample for class *railway*, whereas in Hameln, there is none for *stagnant water*. In level III, there are 25 categories in Hameln and 27 categories in Schleswig. In total, there are 2945 land use objects in Hameln and 4345 in Schleswig.

4.1.2. Test setup: Each test dataset is split into two blocks for cross validation. The block size is 10000 x 15000 pixels (6 km^2) and 30000 x 15000 pixels (18 km^2) for Hameln and Schleswig, respectively. In each test run, one block is used for training and the other one for testing. In each block about 15% samples from all training samples are taken out as validation samples, and the rest is for training. We compare all network variants described in section 3.3. In all cases, the evaluation is based on the number of correctly classified database objects (polygons) and we report the average overall accuracy (OA) and F1 scores over both test runs of cross validation.

level I	level II	level III	#H	#S
settlement	*residential area (residential)*	*residential in use*	528	803
		extended residential area (ext. residential)	34	61
	industry area (industry)	*factory area (factory)*	87	39
		business area (business)	193	158
		energy area (energy)	54	62
	mixed-used area (mixed)	*mixed-used area (mixed)*	9	127
		Forestry	-	51
	special area (special)	*special area (special)*	135	-
		public usage	-	143
		historic setup	-	13
	recreation area (recreation)	*sport & leisure area (leisure)*	27	64
		Graveyard	299	365
traffic	*road traffic*	*motor-road*	491	732
		traffic-guided area (traffic-guided)	87	75
	path	*roadway*	244	-
		foot / bike path	233	-
		Path	-	287
	parking lot (parking)	*parking lot (parking)*	91	76
	railway	*railway*	39	-
		railway-guided area (rail.guided)	47	-
vegetation	*agriculture*	*farm land*	58	214
		grass land	-	427
		garden land	83	13
		fallow land	17	-
	forest	*hardwood*	-	117
		Softwood	-	37
		hard or softwood	33	-
		hard & softwood	15	134
	grove	*Grove*	51	88
	undeveloped	*Undeveloped*	31	-
		moor or swamp	-	101
		vegetation free area (non-veg.)	-	15
water system	*flowing water (flowing)*	*River*	19	29
		Creek	40	12
	stagnant water (stagnant)	*stagnant water (stagnant)*	-	102

Table 1. Hierarchical class structure. Abbreviations are shown in brackets. *#H / #S*: number of samples in level III for Hameln and Schleswig, respectively. “-“ indicates that a class does not occur in the respective dataset.

To obtain the land cover input, the *FuseEnc* network of Yang et al. (2019) is applied, where RGB, IR and nDSM data serve are used. It was trained like in the original publication, where pixel-based overall accuracies of 89.1% and 87.3% were reported for Hameln and Schleswig, respectively. We differentiated eight land cover classes (*building*, *sealed area*, *bare soil*, *grass*, *tree*, *water*, *car* and *others*), so that the input patches for the networks for predicting LU have 14 bands (4 DOP bands, nDSM, binary mask, 8 land cover inputs).

For the training of all network variants, the hyper-parameter of the focal loss (eq. 2) is set to γ = 1; the hyper-parameter for weight decay is 0.0005. We train all networks for eight epochs (an epoch consists of a set of iterations so that in one epoch all samples are used for training once. The number of iterations per epoch is the number of training samples divided by the mini batch size), using a base learning rate of 0.001 and reducing it to 0.0001 after four epochs. The mini batch size is set to 12. We apply data augmentation by vertical and horizontal flipping and by applying random rotations in certain intervals, where the interval and, thus, the amount of data augmentation depends the size of the polygons. When *tiling* is applied, the interval is 30° for polygons that have to be split because they do not fit into the input window of the CNN and 5° for all the other polygons. Consequently, after data augmentation, there are 354178 and 479978 patches for Hameln and Schleswig, respectively.

4.2 Evaluation

Tab. 2 presents the results of the land use classification of all network variants in the two test sites. In section 4.2.1, we compare the results of the three network variants described in section 3.3. After that, we take an exemplary closer look at the performance of one of the better variants (*HierLuNet-F2C*) in section 4.2.2.

4.2.1 Comparison between the network variants: Comparing the network variants described in section 3.3, the multi-task learning (*LuNet-MT*) delivers better results in terms of OA in most cases in both test sites. First, we compare the two network architectures of multi-task learning (*LuNet-MT*) and its variant with hierarchical training and inference in a *coarse-to-fine* manner (*HierLuNet-C2F*). In both sites, *LuNet-MT* performs better than *HierLuNet-C2F* in all evaluation metrics of all category levels. In Hameln, compared to *LuNet-MT,* the drops of *HierLuNet-C2F* in terms of OAs are around 2.5% in level II and level III, whereas the OAs of level I are very similar close (-0.2%). Besides, there are larger drops in terms of average F1 scores in level II and III, which are around 4%. However, the results of *HierLuNet-C2F* in Schleswig are much worse than the ones of *LuNet-MT*: the drops in terms of OA are 3.5% (I), 4.2% (II) and 6.0% (III), whereas the drops in terms of average F1 score are 5.2% (I), 5.1% (II) and 4.9% (III). Like in Hameln, the drops of average F1 scores are a little larger than the ones of OAs. Second, we compare *LuNet-MT* with *HierLuNet-F2C*, the one with hierarchical training and inference in a *fine-to-coarse* manner. In Hameln, the OA of *LuNet-MT* outperforms the one of *HierLuNet-F2C* up to 1.8% over all levels. The difference in terms of average F1 score is much larger (5.4% at level II and 3.0% at level III). Nonetheless, there is an exception for the mean F1 score at level I where there is an increase of 1.2% in *HierLuNet-F2C*. Looking at the results in Schleswig, there is another picture in terms of OA: *HierLuNet-F2C* outperforms *LuNet-MT* by 2.5% at level II and 1.3% at level III, but with a drop of 0.4% at level I. There is a drop of average F1 scores with 1.9% at level I, but at the level II we find an improvement of 0.6% whereas at level III the average F1 scores are most similar. In conclusion, *HierLuNet-F2C* performs almost equivalent as *LuNet-MT* in Schleswig. The final comparison is between *HierLuNet-F2C* and *HierLuNet-C2F*, where in Schleswig the former outperforms the latter in terms of OA and average F1 score over all levels, and the largest difference of OA is the one at level III with 7.3%. In Hameln, *HierLuNet-F2C* delivers mostly better results except for the average F1 score at level II for which there is a drop of 1.6%. Thus, it seems that the hierarchical LU classification benefits more from a fine-to-coarse procedure.

Over the three variants, it is clear that the multi-task learning (*LuNet-MT*) delivers better results in most cases. The big disadvantage of *LuNet-MT*, however, lies in the fact that their predictions do not guarantee a consistent hierarchical result. For instance, in Hameln, 9.1% of the predictions are non-consistent with the hierarchy, whereas in Schleswig the amount is 15.1%. These predictions are obviously not suitable for further processing. On the other hand, the drawback of *HierLuNet-C2F* and *HierLuNet-F2C* is that if the first prediction is wrong (level I in the former and level III in the latter), the successive predictions in the finer (coarser) levels would be wrong as well.

Network variant	**Category level**					
	I		**II**		**III**	
	OA	$\overline{F1}$	OA	$\overline{F1}$	OA	$\overline{F1}$
Hameln						
LuNet-MT	**90.8**	82.9	**73.4**	**58.0**	**64.9**	**44.0**
HierLuNet-C2F	90.6	82.9	71.2	54.2	62.2	40.3
HierLuNet-F2C	90.5	**84.1**	71.8	52.6	63.1	41.0
Schleswig						
LuNet-MT	**88.1**	**83.4**	67.6	53.7	62.5	**41.5**
HierLuNet-C2F	85.6	78.2	63.2	48.6	56.5	36.6
HierLuNet-F2C	87.7	81.5	**70.1**	**54.3**	**63.8**	41.3

Table 2: Overview of the results of hierarchical land use classification for all network variants (cf. section 3.4.1) for Hameln and Schleswig. $\overline{F1}$: average F1 score [%], OA: Overall Accuracy [%]. Best scores are shown in bold font.

Comparing the results achieved by all variants, the expected decrease of classification accuracy when increasing the semantic resolution is obvious. At the coarsest level (I), the OA is around 90% for all variants. It would seem that CNN-based classification at this level is better than the one of the CRF-based method (85%) reported in (Albert et al., 2016), although the class structures are not identical and, thus, a direct comparison is impossible. At the intermediate level, we observe a drop in OA of about 15%-20%. The fact that the drop in the average F1 scores is even larger indicates that a non-negligible number of classes can no longer be differentiated. Finally, the performance at the finest level is even lower, with a drop in the order of another 5%-10% in OA compared to level II. Again, the drop in the average F1 scores is larger. There are two main reasons for the problems at the semantic level II. First, the number of training samples of individual classes is much lower, leading to insufficient representation of this category (cf. Tab. 1). Second, in many cases, the properties of the objects in shape and composition of land cover types are quite similar among classes derived from the same ancestor category. For instance, class *industry area* in level II is very similar to *residential area* with dense buildings and sealed streets.

4.2.2 Detailed analysis of *HierLuNet-F2C*: Tab. 3 presents the F1 scores and OA for all classes achieved by this network variant, which applies hierarchical training and inference in a fine-to-coarse manner. We analyse these results level by level.

Level I: In this level, the four categories can be separated easily in both Hameln and Schleswig. However, in both cases, average F1 scores of less than 80% for the class *water system* indicate a problem with that class. This may partly be due to the fact that there are very few samples of that class (2.0% of all objects in Hameln and 3.3% in Schleswig). Furthermore, an analysis of the confusion matrix shows that about 30% of the samples of *water system* are confused with *traffic* in both sites. The reason could be that both kinds of object are very similar in shape and land cover components (e.g. both are surrounded by grass and trees, and they may be occluded by the latter), which, in combination with the lack of training samples for water, prevents the CNN from learning to differentiate these classes.

Level II: the categories of level II are related to level I based on the semantic relationships shown in Tab. 1. We analyse the results according to the categories of level I.

There are only three level II sub-categories of ***settlement*** achieving F1 scores over 50% in both data sets (*residential area*, *industry area*, *recreation area*). Samples of the other categories are very hard to be correctly recognized. The main source of

Hameln					
level I		**level II**		**level III**	
category	**F1**	**category**	**F1**	**category**	**F1**
settlement	**91.7**	*residential*	**83.8**	*residential in use*	**85.2**
				ext. residential	**57.9**
		industry	**58.5**	*factory*	25.6
				business	48.7
				energy	27.6
		mixed	0	*mixed*	0
		special	33.0	*special*	33.0
		recreation	**73.4**	*leisure*	18.6
				graveyard	**73.7**
traffic	**92.5**	*road traffic*	**82.1**	*motor-road*	**86.4**
				traffic-guided	**53.5**
		path	**78.1**	*guideway*	**55.2**
				foot / bike path	**50.7**
		parking	38.7	*parking*	38.7
		railway	**51.6**	*railway*	45.3
				rail. guided	**53.5**
vegetation	**80.0**	*agriculture*	**72.8**	*farm land*	**54.4**
				garden land	**57.8**
				fallow land	0
		forest	**54.1**	*hard or softwood*	47.8
				hard & softwood	0
		grove	32.3	*grove*	32.3
		undeveloped	6.0	*undeveloped*	6.0
water system	**72.2**	*water system*	**72.2**	*river*	0
				creek	**72.1**

Schleswig					
level I		**level II**		**level III**	
category	**F1**	**category**	**F1**	**category**	**F1**
settlement	**90.4**	*residential*	**80.0**	*residential in use*	**81.9**
				ext. residential	**58.9**
		industry	**52.8**	*factory*	6.0
				business	43.3
				energy	29.1
		mixed	26.8	*mix.res*	23.5
				forestry	24.6
		special	29.6	*public usage*	31.3
				historic setup	0
		recreation	**62.5**	*leisure*	38.3
				graveyard	**58.9**
traffic	**86.1**	*road traffic*	**81.0**	*motor-road*	**84.4**
				traffic-guided	41.6
		path	**65.1**	*path*	**65.1**
		parking	3.7	*parking*	3.7
vegetation	**90.6**	*agriculture*	**89.9**	*farm land*	**84.0**
				grass land	**80.8**
				garden land	34.0
		forest	**84.5**	*hardwood*	49.4
				softwood	**53.7**
				hard & softwood	22.6
		grove	43.9	*grove*	43.9
		undeveloped	46.9	*moor or swamp*	48.4
				non-veg.	15.6
water system	**58.8**	*flowing*	29.8	*river*	29.7
				creek	0
		stagnant	**63.8**	*stagnant*	**63.8**

Table 3: F1 scores (%) of individual category of all levels from *HierLuNet-F2C*. The F1 scores over 50% are printed in bold font.

errors is a confusion between *mixed-used area* and *industry area*. Again, this may be due to their similar appearance and compositions of land cover (cf. Fig. 5-a).

Among the sub-categories of ***traffic***, the *road traffic* and *path* are differentiated most easily (F1 scores > 65% in both sites). *Parking lot* is classified much better in Hameln than in Schleswig. It is most frequently confused with *road traffic* and *industry area*; in Schleswig, about 34% and 39% of the *parking lot* objects are classified as *road traffic* and *industry area*, respectively. This may be attributed to the similar appearance of these objects. Fig. 5-a shows an example for a confusion between *parking lot* and *industry area*.

Among the sub-categories of ***vegetation***, *agriculture* is particularly well classified (F1 > 70%) in both cases. In Schleswig, *forest* also achieves a high F1 score (84.5%), while there are problems in Hameln, where much fewer samples of that class are available (48, as opposed to 288 in Schleswig). The other sub-categories are not differentiated very well. The largest amount of confusion for *grove* occurs with *recreation area* and *forest*. These classes mostly consist of low and high vegetation, which makes them very similar to *grove* (cf. Fig. 5-b). The category *undeveloped* is mainly confused with *agriculture*.

Level III: while in level III, some classes can be differentiated very well, e.g. *residential in use* or *motor-road*, in general it is more difficult to separate them than those of the other levels. More than half of the categories achieve F1 scores smaller than 50%. Again, a major reason is that the number of training samples for some class is quite small.

In summary, as the number of categories increases from level to level, they are harder to be classified correctly. While at the finer levels, the similarity in appearance and land cover composition of some categories (e.g. *industry area* vs. *mix-used area*; *grove* vs. *forest*) may be problematic under all circumstances, it would seem obvious that in order to achieve satisfactory results, the number of training samples has to be increased. Given the fact that the number of objects is given by the database, the way to do so is to increase the size of the area that is processed.

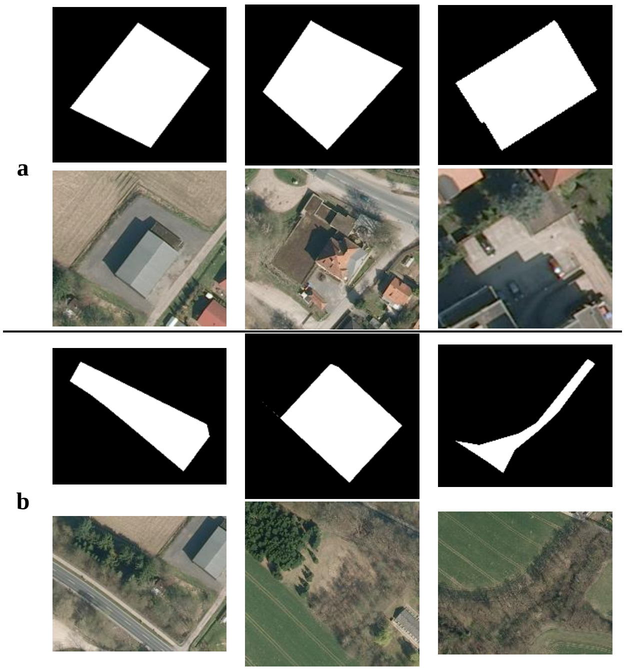


Figure 5: Similar land use objects in category level II with polygon masks (binary images) and DOP (RGB). From left to right in **group a**: *mixed used area, industry area, parking lot*; From left to right in **group b**: *recreation area, grove, forest.* The images are rescaled for visualization.

5. CONCLUSION

In this paper, we have presented three CNN-based methods for the classification of LU in multiple hierarchal semantic levels. The first CNN classifies the categories of all levels independently, while the other two apply the hierarchical training and inference (coarse to fine vs. fine to coarse) in a manner that guarantees hierarchical consistency. All methods require a strategy for providing the CNN with an input of an appropriate size. The categories at the coarsest level are most easily to be discerned: in both test sites, we achieved an OA around 90%. As the number of categories is increased, they are harder to be classified correctly. The main reasons seem to be that the number of training samples per class is heavily reduced and at the finer levels, there are more and more categories that have very similar appearance. Our experimental results also show that multi-task learning without applying hierarchical training and inference delivers good results in most cases, yet suffering from severe hierarchical inconsistency. For instance, there are 15.1% non-hierarchical predictions in Schleswig. By introducing fine-to-coarse hierarchical training and inference into the CNN, the hierarchical predictions are guaranteed and the difference in terms of OA to the predictions of multi-task learning are less than 2% over all levels in both test sites, which is quite satisfactory.

In the current results we have observed some overfitting when comparing the classification results on the training and the validation data set, which we will further investigate in the future by simplifying the network (and thus reducing the number of parameters to be learnt) and by increasing the amount of training data. In our future work, we want to improve the prediction procedure so that we obtain the most probable tuple of class labels that is consistent with the class hierarchy for every object rather than fixing the class label at the coarsest or the finest level of the hierarchy as it is done now in the C2F and F2C strategies. Second, similarly to (Albert et al., 2016) we will further analyse the class structures used for the classification based on the object catalogue. Finally, an increase of the number of training samples, which requires the availability of true annotations for larger areas, is a pre-requisite for reliable results (Kaiser et al., 2017). Such samples can be derived automatically from existing maps if a strategy to mitigate errors in the class labels of training samples (label noise) is developed, e.g. (Maas, et al., 2019).

ACKNOWLEDGEMENT

We thank the Landesamt für Geoinformation und Landesvermessung Niedersachsen (LGLN), the Landesamt für Vermessung und Geoinformation Schleswig Holstein (LVermGeo) and Landesamt für innere Verwaltung Mecklenburg-Vorpommern (LaiV-MV) for providing the test data and for their support of this project.

REFERENCES

Abadi, et al., 2015. Large-scale machine learning on heterogeneous systems. https://www.tensorflow.org (accessed 09/01/2019).

Albert, L., Rottensteiner, F., Heipke, C., 2016. Contextual land use classification: how detailed can the class structure be? ISPRS Archives of Phot., Rem. Sens. and Spat. Info. Sc. Vol. XLI-B4, 11-18.

Albert, L., Rottensteiner, F., Heipke, C., 2017. A higher order conditional random field model for simultaneous classification of land cover and land use. ISPRS JPhRS 130: 63-80.

Arbeitsgemeinschaft der Vermessungsverwaltungen der Länder der Bundesrepublik Deutschland (AdV), 2008. ALKIS®-Objektartenkatalog 6.0. Available online (accessed 27/01/2020): http://www.adv-online.de/GeoInfoDok/GeoInfoDok-6.0/Dokumente/

Barnsley, M. J. & Barr, S. L., 1996. Inferring urban land use from satellite sensor images using kernel-based spatial reclassification. Photogrammetric Engineering & Remote Sens. 62(8): 949–958.

Deng, J., Ding, N., Jia., Y., Frome, A., Murphy, K., Bengio, S., Li., Y., Neven, H., Adam, H., 2014: Large-scale object classification using label relation graphs. European conference on Computer Vision (ECCV), Lecture Notes in Computer Science, Vol. 8689, Springer, Cham, pp. 48-64.

Gerke, M., Heipke, C., 2008: Image-based quality assessment of road databases. International Journal of Geographical Information Science. Vol. 22, pp. 871-894

Guo, Y., Liu, Y., Bakker, E.M. et al., 2018: CNN-RNN: a large scale hierarchical image classification framework. Multimedia Tools and Applications 77, pp. 10251-10271

Helmholz, P., Rottensteiner, F., Heipke, C., 2014. Semi-automatic verification of cropland and grassland using very high resolution mono-temporal satellite images. ISPRS JPhRS 97: 204-218.

Hermosilla, T., Ruiz, L. A., Recio, J. A., Cambra-López, M., 2012. Assessing contextual descriptive features for plot-based classification of urban areas. Landscape and Urban Planning, 106(1): 124-137.

Hua, Y., Mou, L., Zhu, X.X., 2019: Recurrently exploring class-wise attention in a hybrid convolutional and bidirectional LSTM network for multi-label aerial image classification. ISPRS JPhRS 149: 188-199.

Hu, H., Zhou, G.T., Dong Z., Liao Z., Mori G., 2016: Learning structured inference neural networks with label relationships. IEEE Conf. CVPR, pp. 2960-2968.

Ioffe, S., Szegedy, C., 2015. Batch Normalization: accelerating deep network training by reducing internal covariate shift. International Conference on Machine Learning, pp. 448-456.

Kaiser, P., Wegner, J.D., Lucchi, A., Jaggi, M., Hofmann, T., Schindler, K., 2017: Learning aeiral image segmentation from online maps. IEEE T-GRS. Vol. 55, pp. 6054-6068.

Krizhevsky, A., Sutskever, I., Hinton, G. E., 2012. ImageNet classification with deep convolutional neural networks. Advances in Neural Information Processing Systems 25 (NIPS'12), Vol. 1, pp. 1097-1105.

Leiva-Murillo, J.M., Gomez-Chova, L., Camps-Valls, G., 2013. Multi-task remote sensing classification. IEEE T-GRS, 51(1), pp. 151-161.

Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollar, P., 2017b. Focal loss for dense object detection. IEEE International Conference on Computer Vision (ICCV), pp. 2999-3007.

Maas, A., Rottensteiner, F., Heipke, C., 2019. A label noise tolerant random forest for the classification of remote sensing data based on outdated maps for training. Computer Vision and Image Understanding, ISSN:1077-3142, Vol. 188, Page: 102782

Montanges, A.P., Moser, G., Taubenböck, H., Wurm, M., Tuia, D., 2015. Classification of urban structural types with multisource data and structured models. IEEE Joint Urban Remote Sensing Event (JURSE), pp. 1–4.

Novack, T., Stilla, U., 2015. Discrimination of urban settlement types based on space-borne SAR datasets and a conditional random fields model. ISPRS Annals of the Phot., Rem. Sens. and Spat. Info. Sc. II-3/W4, pp. 143–148.

Yang, C., Rottensteiner, F., Heipke, C., 2018: Classification of land cover and land use based on convolutional neural networks.

ISPRS Annals of Phot., Rem. Sens. and Spat. Info. Sc. Vol. IV-3, pp. 251-258

Yang, C., Rottensteiner, F., Heipke, C., 2019: Towards better classification of land cover and land use based on convolutional neural networks. ISPRS Archives of Phot., Rem. Sens. and Spat. Info. Sc. Vol. XLII-2/W13, pp. 139-146

Walde, I., Hese, S., Berger, C., Schmullius, C., 2014. From land cover-graphs to urban structure types. International Journal of Geographical Information Science 28(3): 584–609.

Zhang, C., Sargent, I., Pan, X., Li, H., Gardiner, A., Hare, J., Atkinson, P.M., 2018. An object-based convolutional neural networks (OCNN) for urban land use classification. Remote Sensing of Environment 216: 57-70.

Zhang, C., Sargent, I., Pan, X., Li, H., Gardiner, A., Hare, J., Atkinson, P.M., 2019. Joint deep learning for land cover and land use classification. Remote Sensing of Environment 221: 173-187.

Zhu, X. X., Tuia, D., Mou, L., Xia, G.-S., Zhang, L., Xu, F., Fraundorfer, F., 2017. Deep learning in remote sensing: A comprehensive review and list of resources. IEEE Geoscience and Remote Sensing Magazine 5(4): 8-36.

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
https://doi.org/10.5194/isprs-annals-V-2-2020-599-2020 | © Authors 2020. CC BY 4.0 License.