TOWARDS BETTER CLASSIFICATION OF LAND COVER AND LAND USE 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: Land use classification, CNN, geospatial land use database, aerial imagery, semantic segmentation

ABSTRACT:

Land use and land cover are two important variables in remote sensing. Commonly, the information of land use is stored in geospatial databases. In order to update such databases, we present a new approach to determine the land cover and to classify land use objects using convolutional neural networks (CNN). High-resolution aerial images and derived data such as digital surface models serve as input. An encoder-decoder based CNN is used for land cover classification. We found a composite including the infrared band and height data to outperform RGB images in land cover classification. We also propose a CNN-based methodology for the prediction of land use label from the geospatial databases, where we use masks representing object shape, the RGB images and the pixel-wise class scores of land cover as input. For this task, we developed a two-branch network where the first branch considers the whole area of an image, while the second branch focuses on a smaller relevant area. We evaluated our methods using two sites and achieved an overall accuracy of up to 89.6% and 81.7% for land cover and land use, respectively. We also tested our methods for land cover classification using the Vaihingen dataset of the ISPRS 2D semantic labelling challenge and achieved an overall accuracy of 90.7%.

1. INTRODUCTION

The goal of *land cover* classification is to assign a class label for each image pixel so that the physical material of its surface (e.g. *grass, asphalt*) is identified. In contrast, *land use* describes the socio-economic function of a piece of land (e.g. *residential, agricultural*) which can contain many different land cover elements, while a specific land cover type can be a part of different land use objects. The information about land use 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 class labels indicating the object land use. The goal of *land use* classification is updating the existing database, which is expected to be easier if the results of land cover classification are available. In this paper, we propose a new method for the classification of land cover and land use based on high-resolution aerial imagery and derived data such as a Digital Surface Model (DSM) and a Digital Terrain Model (DTM). Both stages of the classification process are based on convolutional neural networks (CNN).

The pixel-based classification (*semantic segmentation*) of images has been tackled by supervised methods, most recently by CNN such as the fully convolution network (FCN, Long et al., 2014) and encoder-decoder based networks (e.g. Noh et al., 2015, Badrinarayanan et al., 2017). These networks have also been applied for land cover classification based on aerial images (Audebert et al., 2018; Marmanis et al., 2018; Volpi and Tuia, 2018; Sherrah, 2016). One of the main problems of such methods is a precise boundary delineation due to the loss of spatial resolution caused by the pooling layers of the CNN. Strategies for solving that problem include dilated convolutions to avoid pooling (Sherrah, 2016), extracting boundaries explicitly as an additional input (Marmanis et al., 2018) and fusing different data sources (Audebert et al., 2018). Another promising strategy is to use *skip connections*, i.e. upsampling low resolution feature maps and adding high resolution features from the encoder part of the CNN, e.g. (Marmanis et al., 2018). Inspired by Maggiori et al. (2017), we argue that a *skip connection* that can learn the combination of feature maps might improve the delineation of boundaries and, thus, improve the classification performance. Building on the *SegNet* architecture of Yang et al. (2018), we propose extensions that apply *skip connections* between the convolution blocks in the encoder part and corresponding blocks in the decoder part in a learnable way instead of elementwise addition. Furthermore, we propose two different methods for fusing RGB images with near infrared and height data. These extensions should lead to an improvement of the accuracy of *land cover* classification.

For *land use* classification, the biggest challenge is the large variation of polygons in terms of their geometrical extent; for instance, *road* objects are thin and long, whereas *residential* objects may cover both, very large and quite small areas. This poses problems for CNN, because they require a fixed input patch size. Yang et al. (2018) tried to solve this problem by decomposing large polygons into smaller parts. They placed RGB data and land cover labels inside the polygons in fixed-size patches with a black background for classification. We argue that using black background leads to a loss of context information for a database object, while using land cover labels means that one neglects the uncertainties of land cover classification. We propose a representation of a polygon by a combination of its shape in the form of a binary mask while using image data (e.g. RGB) from a patch of fixed size. In addition, we use the class scores from land cover classification as input rather than the class labels. As the size and position of a polygon in a patch to be classified can vary a lot, we propose a two-branch network which uses a focus on the relevant part of the data region of interest, ROI) in addition to the whole input image. Moreover, decomposing polygons into patches implies that the information about polygon shape is partly lost. We propose another method

This contribution has been peer-reviewed.
https://doi.org/10.5194/isprs-archives-XLII-2-W13-139-2019 | © Authors 2019. CC BY 4.0 License.

to convert irregular polygons to a fixed size fitting the input of the CNN by rescaling, so that shape information is preserved. The scientific contributions of this paper can be summarized as follows:

- We propose networks incorporating learnable skip connections for land cover classification to achieve a better representation of the object boundaries in the results.
- We propose new fusion frameworks for combining RGB images with infrared and height data, analysing the effects of the fusion on the results of land cover classification.
- We proposed a CNN-based method for land use classification that improves existing work by using land cover posteriors instead of class labels as input and by applying a two-branch classification network, zooming in at the relevant part of the data (ROI) in addition to using a fixed-sized image patch.
- We apply an additional method for bringing the polygons to the input size of the CNN by rescaling, so that the shapes of large polygons are preserved, while small objects will cover a larger portion of the CNN input.

For both tasks, we conduct experiments using two test sites and compare the results to show the benefits and the remaining problems of the proposed methods. Land cover classification is also applied to the Vaihingen dataset of the ISPRS 2D semantic labelling challenge for a comparison to the state of the art.

In section 2, we give a review of related work. Our approaches for land cover and land use classification are presented in sections 3 and 4, respectively. Section 5 describes the experimental evaluation of our approach. Conclusions and an outlook are given in section 6.

2. RELATED WORK

Land cover classification implies the prediction of dense class labels for input images. There are quite a few CNN-based approaches to achieve this goal; a recent overview for remote sensing applications is given in (Zhu et al., 2017). One strategy is to apply networks which can directly deliver predictions at pixel level, e.g. FCN (Long et al., 2014) or encoder-decoder based networks (Noh et al., 2015). A FCN applies convolution and pooling operations to the input image, leading to a map of signals having lower spatial resolution. After that, the signal is up-sampled directly to the full resolution of the input to make class predictions. Encoder-decoder networks apply convolution and pooling operations in the same way as standard CNN in the encoder part. After that, upsampling is carried out in a decoder network that is structured symmetrically to the encoder in order to obtain predictions at the resolution of the input image. A similar strategy is pursued by *SegNet* (Badrinarayanan et al., 2017) and *U-Net* (Ronneberger et al., 2015), applying end-to-end learning of all parameters, including those of the decoder part. Pooling operations do not only make the learned features invariant to image transformations, but they also enlarge the receptive field to incorporate more context information in an implicit way. However, they lead to a loss of spatial resolution and, consequently, to inaccurate object boundaries. To mitigate this problem, many authors use skip connections that directly connect feature maps of low levels to high levels in a network, e.g. (Long et al., 2014; Zhao et al., 2017; Lin et al., 2017a), typically inserting them just before the classification layer. Variants of such networks have been used for land cover classification, achieving promising results. Marmanis et al. (2018) extract edge maps from images by applying a Holistically-Nested Edge Detection (HED) framework (Xie et al., 2017). The edge maps are concatenated with images as input for FCN and *SegNet*, and the outputs are combined for the final class prediction. Good results are achieved at the cost of many training stages and a huge number of parameters. Audebert et al. (2018) investigate *SegNet* and *ResNet* (He et el., 2016) and the integration of multispectral and height information in one model, and achieve promising results. Both methods just cited use skip connections by a simple elementwise addition of feature maps (Long et al., 2014), so that the combination of the features of different resolution cannot be learned. Learning feature combinations in skip connections was proposed by Maggiori et al. (2017). They concatenate feature maps of different resolutions and then convolve the concatenated maps with 1 x 1 filters, thus reducing the dimension of the feature maps. Volpi and Tuia (2018) proposed a network which adopts such learnable feature combinations and learns class boundaries explicitly, achieving accurate results. However, all methods cited so far only introduce skip connections before the classification layer. In a symmetric encoder-decoder structure, we can utilize the feature maps of the encoder part to enrich the representation in the decoder part. For instance, in *U-Net*, originally designed for biomedical applications (Ronneberger et al., 2015), the skip connections are introduced between the last convolutional layers in corresponding encoder and decoder convolution blocks symmetrically, concatenating the feature maps for further processing. Here, we combine the ideas of Ronneberger et al. (2015) and Maggiori et al. (2017). We build a structure similar to *U-Net*, but concatenating the outputs of all convolutional layers at each resolution and using 1 x 1 convolutions to learn the combination of encoder and decoder features.

Existing methods for land use classification differ by the data sources, the primitives to be classified, the features used for classification and the classifiers used to predict the class labels (Albert et al., 2017). Most approaches solve the problem in two steps: first, they determine land cover, and then they use land cover to support the land use classification (Hermosilla et al., 2012). Typically, *hand-crafted features* derived from image data or land cover are applied in this context. Examples for features taking into account land cover are *spatial* and *graph-based-metrics*. For instance, such 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). These features are then delivered to supervised classifiers such as Random Forests (Albert et al., 2017) or Support Vector Machines (SVM). Contextual models like Conditional Random Fields (CRF) have also been applied (Albert et al., 2017). In the context of CNN, the classification of land use objects from a geospatial database shares some resemblance to object detection. The main difference is that in object detection, interesting regions need to be determined automatically before presenting them to a CNN for classification (Ren et al., 2015). In our case, we know the locations and shapes of land use objects, yet their variations of shapes are very large. The first work classifying land use objects from a geospatial database by CNN is (Yang et al., 2018). They decompose large polygons into multiple patches suitable for being classified by a CNN. However, they used a pre-defined black image and only put the image data (RGB and land cover labels) into that image, which leads to a loss of context information. As pointed out earlier, the use of land cover label means that the uncertainty of the land cover, which may be very essential for the correct classification of polygons, is not considered. Building on (Yang et al., 2018), in this paper we present an improved method that tackles these problems in the way already indicated in Section 1.

3. CNN-BASED CLASSIFICATION OF LAND COVER

Our CNN for land cover classification is based on SegNet (Badrinarayanan et al., 2017). Compared to FCN, SegNet delivers improved dense pixel predictions and requires lower computational costs (Audebert et al., 2018). Section 3.1 outlines our network, referred to as *SkipNet*. Section 3.2 presents some network variants.

3.1 SkipNet

Like *SegNet*, *SkipNet* (Fig. 1) applies a symmetric encoder-decoder structure. The input size is 256 x 256 pixels with three bands. There are four blocks in the encoder part, each consisting of three convolutional layers followed by batch normalization (BN; Ioffe et al., 2015) and a rectified linear unit (ReLU) for non-linearity. At the end of the block, there is a max-pooling layer. Symmetrically, the decoder part consists of four blocks, each starting with an upsampling layer that applies bilinear interpolation, followed by three convolutional layers, batch normalization and a ReLU unit. The filter size of each convolution is 3 x 3. We expand this architecture by skip connections, using the mechanism shown in Fig. 2. Similar to *U-Net*, we first concatenate features from the encoder and the decoder parts, and then use learned 1 x 1 convolutions to obtain the combined feature map. Finally, to predict the class labels at the resolution of the input image, there is a 1 x 1 convolutional layer converting the output of the previous layer to a vector of Mclass scores for each of the $H \times W$ pixels of the input image. For each pixel *i* of the image to be classified, this results in a vector $\boldsymbol{z}_{\boldsymbol{LC}}^{\boldsymbol{i}} = (z_{LC^1}^i, \dots, z_{LC^M}^i)^T$ of class scores, where $\mathbb{C}_{LC} = \{C_{LC^1}, \dots, C_{LC^M}\}$ is the set of land cover classes and $z_{LC^c}^i$ is the class score for class C_{LC^c}. These class scores are normalised by a softmax function delivering the posterior probability $P_i(C_{LC^c}|x)$ for pixel *i* to take class label C_{LC^c} given the image data *x*:

$$P_i(C_{LC^c}|x) = \text{softmax}\left(\boldsymbol{z}_{\boldsymbol{LC}}^{\boldsymbol{i}}, \mathrm{C}_{\mathrm{LC}^c}\right) = \frac{exp(z_{LC^c}^i)}{\sum_{l=1}^{M} exp(z_{LC^l}^i)}. \quad (1)$$

All parameters of convolutional layers are learned during in the training process, which is based on stochastic mini-batch gradient descent (SGD) using backpropagation for computing the gradients. We apply a variant of the focal loss (Lin et al., 2017b) as the objective function. As the original focal loss is designed for binary classification, we extend it to be suitable for multiclass problems, referring to it as the *extended focal loss*:

$$L = -\frac{1}{W \cdot H \cdot N} \sum_{c,i,k} \left[y_{LC^c}^{ik} \cdot (1 - P_i(C_{LC^c}|X_k))^{\gamma} \cdot log\left(P_i(C_{LC^c}|X_k)\right) \right] \quad (2)$$

where *k* is the index of an image, *X*$_k$ is the *k*th image in the mini-batch and *N* is the number of images in a mini-batch. The indicator variable $y_{LC^c}^{ik}$ is 1 if the training label of pixel *i* in image *k* is identical to C_{LC^c} and 0 otherwise, and γ is hyper parameter (set to 1 in our experiments). The sum in (2) is taken over all potential class labels for all pixels of all images of a mini-batch. Compared to the standard cross-entropy loss function, we found this formulation of the loss not only to deliver better predictions but also to accelerate our training procedure. In the training procedure, we also applied weight decay with 0.0005, a step learning policy and used a mini-batch size of 4. The learning rate was set to 0.01 and decreased to 0.001 after 30 epochs in a total of 50 epochs training.

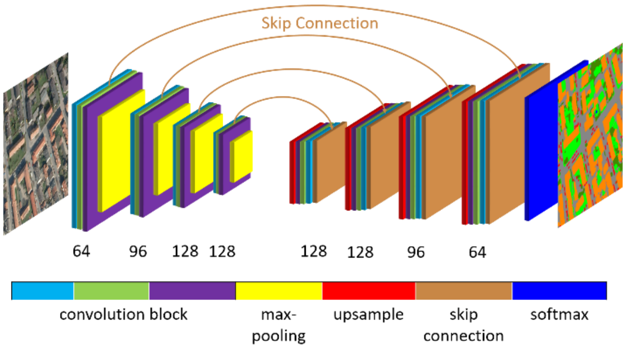

Figure 1. The architecture of *SkipNet*.

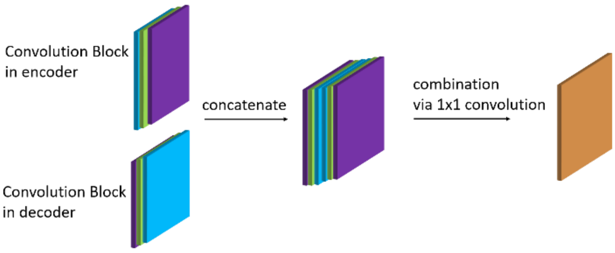
Convolution Block in encoder
Convolution Block in decoder
concatenate
combination via 1x1 convolution

Figure 2. Structure of a skip connection. Colour code: cf. Fig. 1.

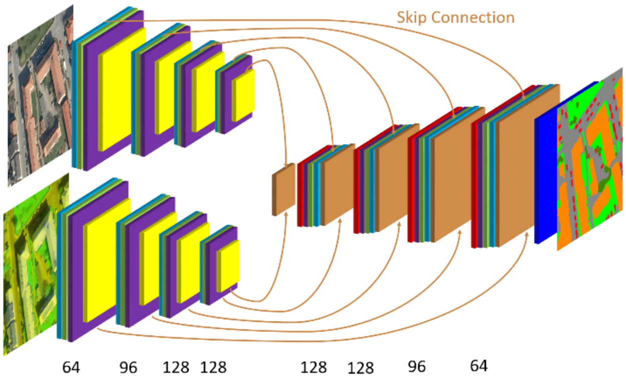

Figure 3. Architecture of *FuseEnc*. Colour code: cf. Fig. 1.

3.2 Network variants

Based on *SkipNet*, we developed four additional variants of the network to incorporate different data sources into one model. The training procedures are identical to the one of *SkipNet*.

First, variant *NoSkip* is similar to *SkipNet* but does not contain any skip connections. We use it to validate the effectiveness of skip connections. In order to validate the effect of the proposed learnable skip connections, we provide the variant *AddSkip*, which is identical to *SkipNet* except that it uses element-wise summation of features rather than the structure shown in Fig. 2 for combining the features.

In order to fuse different data sources, Sherrah (2016), Audebert et al. (2018) and Yang et al. (2018) apply separate network branches to extract features from different image sources and then fuse them by concatenation directly before the softmax layer. To set up a baseline of fusion, variant *FuseDec* applies fusion in a similar way, concatenating the feature vectors of the output of the convolutional layers from the last convolutional block and just adding a 1x1 convolution layer for fusion before the softmax layer. As *FuseDec* requires twice as many parameters as *SkipNet* and because we believe that the encoder already delivers high-level features encoding a good image representation, we propose another network variant for fusion (Fig. 3). This variant, referred to as *FuseEnc*, fuses the features at the end of the encoder: two separate encoder branches are applied to extract features from different data sources and a united decoder is used to upsample the fused features. At each level of the decoder, skip connections from both encoder networks are combined with the decoder output, again using 1 x 1 convolutions to reduce the dimension of the feature vectors.

4. CNN-BASED CLASSIFICATION OF LAND USE

The classification of land use is based on a CNN taking an image patch of 256 x 256 pixels and returning a land use label. As the CNN requires a fixed input size while the land use objects vary considerably in their extent, we start with patch preparation, which is described in Section 4.1. Section 4.2 outlines the basic CNN structure used for classification, while Section 4.3 presents network variants.

4.1 Patch preparation

We propose two different strategies for patch preparation. The first strategy, *cropping*, preserves the original image resolution, which implies that large polygons have to be split into smaller patches. The second strategy, *rescaling*, rescales the polygons so that they fit into the input window of the CNN. Both methods rely on the object boundary polygons from the geospatial database. The polygon shape is represented in the form of a binary object mask, where a value of 255 indicates that a pixel is inside the polygon, whereas pixels outside the polygon are set to 0.

4.1.1 Cropping: Using this strategy, we first check if the polygon fits into a window of 256 x 256 pixels at the resolution of the original data. If this is the case, we place such a window over the polygon so that the polygon centre coincides with the centre of the window (*small* polygon); for polygons that do not fit into a single patch, we split the window enclosing the polygon into a series of tiles of 256 x 256 pixels with an overlap of 50%. (*large* polygon) For each tile, we produce an input image having $N = 4 + N_C$ bands. The first three bands of that image correspond to the RGB data, the fourth band is the binary object mask and the remaining N_C bands correspond to the pixel-wise class scores from land cover classification. The ground sampling distance (GSD) is identical to the one of the input image.

For each tile, we also check the proportion of its area that is inside the database object. If this proportion is smaller than a threshold (set to 0.005% of the area of this tile), the tile will be excluded. As this still leads to a large number of tiles for *large* polygons, we reduce the computational burden, by randomly selecting 40% of the remaining tiles for further processing. Each selected tile results in a patch to be classified; the corresponding *N*-band image is produced by cropping the RGB image, the binary object mask and the class scores to the tile extents while preserving the original resolution. These images are referred to as *Cr-N* images. To compare results based on land cover class scores as opposed to land cover classes, we also generate images having five bands, i.e. the boundary mask, RGB and the land cover labels (*Cr-5* images).

4.1.2 Rescaling: In the case of objects that have to be split in patches, cropping will lead to tiles in which the overall shape of the objects is not preserved well. Thus, we suggest this alternative option where we scale the images (RGB and pixel-wise class scores) and the binary object mask in each axis independently such that the fit into a window of 256 x 256 pixels, resulting in an input having *N* bands as well. The grey values of the images (RGB and pixel-wise class scores) are determined by bilinear interpolation, while for the object mask we use nearest neighbourhood interpolation. These images are named as *Rs-N* images.

4.2 LuNet

This network is based on *LiteNet* (Yang et al., 2018) and consists of four main convolutional blocks at the beginning and two branches towards the end (Fig. 4). Each of the main convolutional blocks consists of three convolution layers followed by BN and ReLU. Each block in the two branches starts with a convolution and a max-pooling layer, followed by a second convolutional layer with ReLU and a final average pooling layer. The filter sizes of all convolution layers are set to 3 x 3, while the number of filters in different blocks is shown in Fig. 4. At the end of the first three convolutional blocks there is a 2 x 2 max pooling layer with stride 2. The upper branch in Fig. 4 starts with a max pooling layer, the aim of which is to extract features that are representative for the entire input image. Due to the variations of the size and position of the polygons inside the images, we propose to use a second branch that just uses a region of interest (ROI) of the input image, i.e. a rectangle aligned with the image grid that tightly encloses the polygon, thus focussing on the most relevant regions in the image. As the size of the ROI varies, we resize the output of the last common convolutional block inside the ROI window to a fixed size of 16 x 16 by bilinear interpolation (Fig. 5). In each branch, an average pooling layer with a window of 8 x 8 is used to determine a 256 dimensional feature vector. Finally, the feature vectors of both branches are concatenated, and the combined feature vector forms the input to a final FC layer that delivers a vector of class scores $\boldsymbol{z_{LU}} = (z_{LU^1}, \ldots, z_{LU^M})^T$, where $\mathbb{C}_{LU} = \{C_{LU^1}, \ldots, C_{LU^M}\}$ is a set of land use classes and z_{LU^c} is the class score of an image in a mini-batch X for class C_{LU^c}. To get a probabilistic class score, the softmax function (eq. 1) is applied to the class scores, thus $P(C_{LU^c}|\mathrm{X}) = \mathrm{softmax}(\boldsymbol{z_{LU}}, C_{LU^c})$. Training is based on mini-batch SGD with weight decay 0.0005, and step learning policy; the function to be optimised is our *extended focal loss*:

$$L = -\frac{1}{N} \cdot \sum_{c,k}\left[y_{LU^c}^k \cdot (1 - P(C_{LU^c}|X_k))^\gamma \cdot log\left(P(C_{LU^c}|X_k)\right)\right] \qquad (3)$$

where X_k is the k^{th} image in the mini-batch, N is the number of images in a mini-batch, and $y_{LU^c}^k$ is 1 if the training label of X_k is C_{LU^c} and 0 otherwise. We set the hyper-parameter γ equal to 1 in our experiments. We train all networks for four epochs, using a base learning rate of 0.001 and reducing it to 0.0001 after two epochs. The mini batch size depends on the network variant and will be given below. In the classification process, the CNN delivers a prediction for each patch. For polygons that had to be split into multiple tiles, we determine the product of the probabilistic class scores of all patches to obtain a combined score for the compound object.

4.3 Network variants

LuNet can be applied to different inputs, and the only adaptation is related to the number of input bands. The variant applied to *Cr-N* images is referred to as *LuNet-Cr-N*, variant *LuNet-Cr-5* uses *Cr-5* images and *LuNet-Rs-N* is based on *Rs-N* images. We also test a variant that consists of an ensemble of *LuNet-Cr-N* and *LuNet-Rs-N*. In this variant, referred to as *LuNet-ENS*, *LuNet-Cr-N* and *LuNet-Rs-N* are applied independently of each other. Subsequently, the probabilistic class scores are multiplied to obtain a final score for the land use label prediction per polygon.

The variant combines the advantages of both methods for generating the input, considering the entire shape via *LuNet-Rs-N* while being based on the original image resolution and considering land use information via *LuNet-Cr-N*. For training these variants, the mini-batch size in training is set to 12. Finally, to validate the effectiveness of the ROI location, we use variant *LuNet-B* that is based on *LuNet-Cr-N* but does not have the ROI branch (bottom branch in Fig. 4). For this variant, the mini-batch size is set to 16, due to its smaller amount of training parameters.

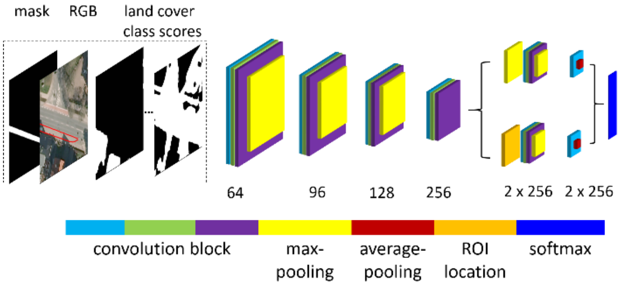

Figure 4. Architecture of *LuNet*.

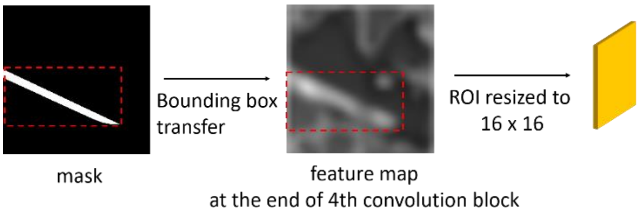

Figure 5. ROI location (red rectangle) and feature extraction.

5. EXPERIMENTS

5.1 Test Data und Test Setup

Our approaches for classification of land cover and land use are evaluated using two test sites located in the cities of Hameln and Schleswig (Germany), covering an area of 2 km x 6 km and 6 km x 6 km, respectively. For each test site, 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. We create both RGB and composite images (COM), the latter ones combining the red and IR band of the DOP with the nDSM to assess the impact of using different inputs for land cover classification.

The references for land cover consist of 37 and 26 manually labelled image patches for Hameln and Schleswig, respectively, each covering 1000 x 1000 pixels (200 m x 200 m). For these datasets, we distinguish 8 land cover classes: *building (build.), sealed area (seal.), bare soil (soil), grass, tree, water, car* and *others*. The reference for land use was derived from the geospatial database. Following investigations by (Albert et al., 2016), we distinguish 10 land use classes: *residential (res.), non-residential (non-res.), urban green (green), traffic (traf.), square, cropland (cropl.), grassland (grassl.), forest, water body (water)* and *others*. To compare our method for land cover classification to other methods, we also apply it to the Vaihingen dataset of the ISPRS 2D semantic labelling challenge, consisting of 33 colour infrared (CIR) images with a GSD of 9 cm. Following the protocol of the benchmark, 16 images with known reference are used for training and the rest (17) for testing. Composite (COM) images are generated based on the nDSM provided by (Gerke, 2015). There are six land cover classes: *impervious surface (imp. surf.), building (build.), low vegetation (low veg.), tree, car* and *clutter* (Wegner et al., 2017). Moreover, all mentioned networks are implemented based on the tensorflow framework (Abadi et al., 2015). We use a GPU (Nvidia TitanX, 12GB) to accelerate training and inference.

5.1.1 Test setup for land cover classification (Hameln and Schleswig): In the tests involving these datasets, we split each image into four non-overlapping tiles of size 500 x 500 pixels, resulting in 148 and 104 tiles for Hameln and Schleswig, respectively. These tiles are randomly split into three groups of equal size for three-fold cross validation. Each tile is split into four overlapping patches corresponding to the input size of the CNN (256 x 256 pixels). In each test run, one group of tiles is used for testing and the others are used for training. In each test, a confusion matrix as well as derived metrics are determined by comparing the results to the reference. We report the average quality metrics over all test runs, focussing on the overall accuracy (OA) and the F1 score, i.e. the harmonic mean of completeness and correctness, all determined on a per-pixel level. In training, we applied data augmentation by flipping all training patches in horizontal and vertical directions and by applying rotations of 90°, 180° and 270° to all patches. We tested two variants of SkipNet that differed by their input: in *SkipNet0*, the input consists only of the RGB images, while in *SkipNet1* we used the composite images (COM). Variants *FuseEnc* and *FuseDec* combine results for both input images (RGB + COM). In the inference procedure, the class labels for a patch of 256 x 256 pixels are predicted six times for the original image and variants that are flipped and rotated just as the training images, multiplying the probabilistic scores to obtain a combined score for classification.

5.1.2 Test setup for land cover classification (Vaihingen): Here, we extract windows of 256 x 256 pixels with an overlap of 128 pixels in both spatial dimensions from the training images, which results in 4426 training patches. We apply the same experiments as for Hameln and Schleswig with the exception that for lack of a blue band in the DOP, we use CIR instead of RGB images. We use this dataset also to validate the impact of our learnable skip connections by applying *NoSkip* and *AddSkip* to CIR images (variants *NoSkip0* and *AddSkip0*, respectively). There are two reference datasets: the *full reference* contains class labels for all pixels, while the *eroded reference* does not consider the pixels near object boundaries (eroded by a circular disc of 3-pixel radius). Here, the quality measures (OA, F1 on a per-pixel level) are determined on the basis of the full reference to make them comparable to the evaluation results for Hameln and Schleswig, but also on the basis of the eroded reference for a comparison to the results of the ISPRS benchmark (Wegner et al., 2017).

5.1.3 Test setup for land use classification: Each test data set 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. Totally, there are 3299 land use objects in Hameln and 4523 in Schleswig. When rescaling is applied for patch preparation, for each database object, the underlying image window is scaled to a size of 256 x 256 pixels and then augmented by horizontal and vertical flipping and by random rotations in steps of about 9°, which results in 141857 and 194489 patches in Hameln and Schleswig respectively. For patches generated by cropping, we differentiate two scenarios. *Large polygons*, i.e. polygons that had to be split because they do not fit into the input window of

Test site	Network	Input	F1 [%]								avg. F1 [%]	OA [%]
			build.	*seal.*	*soil*	*grass*	*tree*	*water*	*car*	*others*		
Hameln	*SkipNet0*	RGB	91.2	83.9	82.5	85.9	87.3	90.9	75.7	41.7	79.9	86.6
	SkipNet1	COM	**94.6**	**87.9**	**85.1**	**88.8**	**89.5**	93.3	76.6	**49.8**	**83.2**	**89.6**
	FuseDec	RGB + COM	93.4	86.7	81.9	87.5	88.0	93.3	**76.8**	43.7	81.4	88.2
	FuseEnc	RGB + COM	94.4	**87.9**	84.0	88.0	88.7	**94.7**	76.2	40.9	81.8	89.1
Schleswig	*SkipNet0*	RGB	87.9	80.2	76.0	78.7	90.5	89.3	69.7	40.4	76.6	84.0
	SkipNet1	COM	91.4	83.1	**82.0**	83.2	90.8	87.3	60.7	36.4	76.9	86.5
	FuseDec	RGB + COM	91.7	84.5	81.6	83.5	90.9	89.4	**72.3**	**43.1**	**79.6**	87.0
	FuseEnc	RGB + COM	**92.4**	**84.8**	80.7	**83.7**	**91.1**	**90.4**	68.6	42.4	79.3	**87.3**
			imp. surf.	*build.*		*low. veg.*	*tree*	*car*		*clutter*		
Vaihingen	*SkipNet0*	CIR	88.5	91.8		79.2	85.8	77.5		**25.0**	74.6	86.3
	SkipNet1	COM	88.9	93.2		80.5	86.7	75.7		16.4	73.6	87.2
	FuseDec	CIR + COM	89.2	93.4		80.9	86.9	79.0		16.4	74.3	87.5
	FuseEnc	CIR + COM	**89.4**	**93.8**		**81.7**	**87.1**	**79.3**		20.5	**75.3**	**87.9**
	*FuseEnc**	CIR + COM	92.0	95.5		85.2	90.2	86.2		22.1	78.5	90.7

Table 1. Results of land cover classification of all networks. COM: composite images, F1: F1 score, OA: Overall Accuracy, both evaluated on the basis of pixels. Best scores are printed in bold font. In Vaihingen, *FuseEnc** is evaluated on the *eroded reference*, other variants are evaluated on the *full reference*.

the CNN, are augmented by horizontal and vertical flipping and by applying random rotations in intervals of 30°. For the other polygons (*small polygons*), we apply horizontal and vertical flipping and random rotations in intervals of 5°. After data augmentation, there are 289020 and 673215 patches in Hameln and Schleswig, respectively. As we have eight land cover classes, the *Cr-N* images have N = 12 bands. We compare the variants *LuNet-B*, *LuNet-Cr-5*, *LuNet-Cr-N*, *LuNet-Rs-N* and the ensemble *LuNet-ENS*, as described in section 4.3. For land use classification, the evaluation is based on the number of correctly classified database objects. Thus, we report OA and F1 scores on a per-object level.

5.2 Evaluation of land cover classification

5.2.1 Evaluation and comparison of network variants: Tab. 1 presents the land cover classification results for all network variants described in section 5.1. In general, the classification works well, with an OA better than 84% and a mean F1 score better than 74% in all cases. All variants have difficulties with underrepresented or heterogeneous classes, in particular *other* (Hameln and Schleswig) and *clutter* (Vaihingen).

Comparing the network variants that are based on the *SkipNet* architecture but use different inputs (*SkipNet0* and *SkipNet1*), we observe an advantage of using the IR and nDSM data (*SkipNet1*) compared to using RGB data (*SkipNet0*), with an improvement of OA between 0.9% (Vaihingen) and 3.0% (Hameln). For Hameln, the F1 scores of all classes increase, leading to an improvement of the average score of +3.3%. In the other areas, there are also improvements in the F1 scores of most classes. This meets our expectation that IR and height help to recognize vegetation and classes sensitive to height (e.g. *building*). However, the improvement of the average F1 score is only small in Schleswig, and it is lower than the one of *SkipNet0* in Vaihingen. This is due to a large drop of F1 for classes having relatively few samples (*car*; *other* / *clutter*). In particular, for moving cars, height information could be expected not to improve classification accuracy; it would seem that this also applies to the heterogeneous classes *other* / *clutter*.

The network variants fusing RGB, IR and height data (*FuseDec*, *FuseEnc*) could be expected to deliver better results than the variants based on a single data source. Tab. 1 shows that this is indeed the case for Schleswig and Vaihingen, where the OA and most F1 scores of both fusion methods are higher than those achieved by *SkipNet1*. For Schleswig, the improvement of OA and the average F1 score is in the order of 1.5-3.0%, for Vaihingen it is somewhat smaller. However, for Hameln, the best result both in terms of OA and F1 is achieved by a network just using the COM images (*SkipNet1)*, only by small margin in OA (0.5%) but by a larger one in F1 (1.4% compared to *FuseEnc*). For Hameln, the additional use of RGB data does not improve the classification quality except for a few classes (*water*, *car*). The reasons are unknown; they might be related to alignment problems between DSM and DOP or to data acquisition under leaf-off conditions. Comparing the two fusion frameworks, *FuseEnc*, which has about 27% fewer learnable parameters, yields slightly better results than the naïve approach of *FuseDec*, with improvements in OA of 0.3% (Hameln and Schleswig) to 0.4% (Vaihingen). In general, the F1 scores are also better, the largest exception being F1 for *car* in Schleswig, which also leads to a slight decrease of the average F1 score for this dataset. Nevertheless, we think that these results show the advantages of sharing the encoder part of the network in the fusion process.

Tab. 1 indicates that there is no clear test winner among the compared methods. In all cases, the best methods use IR and height data. For two datasets, the additional use of RGB data improves the OA, while it decreases OA by a small margin for the third dataset. It would seem that in most cases the fusion framework of *FuseEnc* is a reasonable choice. Finally, we used the pixel-based class scores of *FuseEnc* as input for the land use classification both for Hameln and Schleswig.

5.2.2 Effectiveness of skip connections: Tab. 2 shows the evaluation results achieved for the variants *NoSkip0* and *AddSkip0* on the Vaihingen dataset. The results show that *NoSkip0* performs worse (-3% in OA) than *AddSkip0*, which shows the importance of using skip connections. A comparison of the results of *AddSkip0* to those of *SkipNet0* (based on the same input) in Tab. 1 reveals the advantage of using a learnable skip connection. *SkipNet0* delivers slightly better results in almost all indices, with an improvement of 0.3% in OA.

Network	F1 [%]					OA [%]
	imp. surf.	*build.*	*low. veg.*	*tree*	*Car*	
NoSkip0	84.8	88.6	75.6	83.2	73.2	83.0
AddSkip0	88.1	91.2	79.7	85.4	73.5	86.0

Table 2. Results of land cover classification using different network variants for Vaihingen (*full reference*).

5.2.3 Comparison to the state-of-art: For Vaihingen, we compare our results to those achieved by state-of-art methods.

Network	F1 [%]										avg.	OA
variant	*res.*	*non-res.*	*green*	*traf.*	*square*	*cropl.*	*grassl.*	*forest*	*water*	*others*	F1[%]	[%]
Hameln:												
LuNet-Cr-N	**83.7**	74.4	74.8	92.4	51.6	**81.2**	**55.5**	75.9	**68.5**	56.4	**71.4**	81.3
LuNet-Rs-N	79.8	73.7	70.7	90.4	61.2	69.4	0.00	72.2	48.0	52.4	61.9	77.6
LuNet-ENS	82.9	**75.8**	**75.9**	**93.1**	**61.3**	77.5	36.5	**80.0**	57.0	**58.9**	70.0	**81.7**
Schleswig:												
LuNet-Cr-N	81.0	55.6	57.4	87.7	18.0	87.7	**39.9**	83.0	62.5	36.4	60.9	74.5
LuNet-Rs-N	83.0	56.8	55.9	86.6	20.8	85.7	36.1	77.8	59.6	33.8	59.6	74.5
LuNet-ENS	**84.4**	**60.1**	**62.6**	**89.8**	**25.6**	**88.7**	39.6	**84.7**	**69.3**	**41.5**	**64.6**	**78.0**

Table 3. Results of land use classification. Network variant: cf. section 5.1.3. F1: F1 score, OA: Overall Accuracy, both evaluated on the basis of objects. Best scores are printed in bold font.

Following the convention of the ISPRS benchmark, the eroded reference is used for evaluation. The benchmark website (Wegner et al., 2017) only lists four (out of more than 100) contributions that deliver an OA that is better than the one of our method *FuseEnc*. The OA of *FuseEnc* (90.7% in Table 1) is only 0.9% worse than the best one (HUSTW5; OA = 91.6%); the other methods having a better OA are NLPR3 (91.2%), CASIA2 (91.1%) and BKHN10 (91.0%). We take this as an indication that our method is on par with the current state of the art.

5.3 Evaluation of land use classification

5.3.1 Evaluation and comparison of network variants: Tab. 3 presents the results of the land use classification for different networks in the two test sites. Again, the accuracy values are satisfactory, though they are at a lower level than those achieved in land cover classification. Comparing the two variants of patch preparation (*LuNet-Cr-N* and *LuNet-Rs-N*) shows a clear advantage of the variant based on cropping (*LuNet-Cr-N*), in particular when considering the F1 scores (+9.5%). For Schleswig, the difference is not as pronounced, with the OA being almost identical and an improvement of the average F1 score of 1.3%. Again, underrepresented classes, in particular *grassland* (less than 1% of the samples in Hameln), are most problematic. The ensemble method (*LuNet-ENS*) outperforms both variants relying on a single patch generation strategy in terms of the OA. For Hameln, the improvement over *LuNet-Cr-N* is relatively small (0.4% in OA), and it is contrasted by a drop of mean F1 mainly due to problems in discriminating the underrepresented classes *grassland* and *water body*. For Schleswig, there is a larger improvement of 3.5% in OA and an improvement of about 4% in mean F1. It would seem that in Hameln, using the results of the network based on rescaled patches does not lead to a better representation of the objects. Perhaps this is due to the fact that, rescaling leads to a loss of information about object dimensions. On the other hand, the results of Schleswig show that there are situations in which the two representations convey complementary information to the classifier. While the large differences in the performance for specific classes requires additional investigations, we think that our results show that the ensemble method can give good results with an accuracy in the order of 80% under different circumstances.

5.3.2 Land cover labels vs. land cover scores: we compared the performance achieved when using land cover labels to the one achieved using land cover posteriors as input. The results are presented in Tab. 4 (*LuNet-Cr-N* and *LuNet-Cr-5*). To be able to assess the impact of the input to the network directly, these results are based on the classification results of patches (not objects). In Hameln, using land cover posteriors instead of label images improves the OA and average F1 score by 1.3% and 2.4%, respectively. Although in Schleswig *LuNet-Cr-5* is slightly better than *LuNet-Cr-N*, the differences are very small (0.1% in OA and 0.3% in F1). The comparison remains inconclusive, but on average there seems to be a slightly positive contribution of using class scores.

5.3.3 Effectiveness of ROI location: Tab. 4 also shows the results of variant *LuNet-B*, which does not consider the branch for ROI location in *LuNet*. Both the OA and the average F1 score are lower than those achieved by the other methods (*LuNet-Cr-N* and *LuNet-Cr*-5). The improvement can be up to 4% in OA (Schleswig) but is also noticeable in Hameln. We take this as an indication that the ROI location branch in *LuNet* has a positive impact on the accuracy of land use classification.

Test site	Network	avg. F1 [%]	OA [%]
Hameln	*LuNet-Cr-N*	**69.6**	**80.6**
	LuNet-Cr-5	67.2	79.3
	LuNet-B	68.5	80.5
Schleswig	*LuNet-Cr-N*	53.8	75.9
	LuNet-Cr-5	**54.1**	**76.0**
	LuNet-B	51.0	71.8

Table 4. Results of *LuNet-Cr-N*, *LuNet-Cr-5* and *LuNet-B* based on patches. Number of patches used for evaluation: 289020 (Hameln) / 673215 (Schleswig).

5.3.4 Influence of object size: Tab. 5 shows the OA achieved by *LuNet-ENS* for three different sets of land use objects. The set *small* consists of all objects represented by a single patch in the classification process, whereas the set *large* consists of all objects that were split in the patch generation phase. The table also contains the combined results (*all*), identical to those shown in Tab. 3. Tab. 5 shows that the OA of large objects is considerably better than the one for small objects. In Schleswig the difference is larger than in Hameln, which may be attributed to the low amount of training samples from the small set after data augmentation (21% of the samples compared to 36% in Hameln). We take this analysis as an indication that work on a better classification of small objects is still required.

object	Hameln		Schleswig	
set	#objects	OA [%]	#objects	OA [%]
large	1812	**84.9**	3029	**84.5**
small	1487	77.9	1494	64.7
all	3299	81.7	4523	78.0

Table 5. OA for three different sets of objects based on *LuNet-ENS* in Hameln and Schleswig.

6. CONCLUSION

In this paper, we have proposed networks for land cover classification. We investigated the performance of the improved networks on RGB images and IR and height data, showing that IR and height data lead to much better results. These data allow for a better discrimination of vegetation and objects sensitive to height. We also proposed two fusion networks and found the fusion at the end of the encoder performs better than the fusion at the end of the decoder while requiring fewer parameters. Compared to the CRF-based results of Albert et al. (2017), the OA is improved by 5.9% and 4.8% in Hameln and in Schleswig, respectively. However, the delineation of object boundaries is still not precise, which is a future focus of research.

We have also proposed improved methods for the classification of land use objects based on CNN, by introducing a two-branch network: one branch focusses on the entire image to extract a global representation and the other one on a smaller relevant area (ROI). The results are very promising, in particular for large objects. We have shown that integrating the information about object shapes by combining two different pre-processing strategies improves land use classification further. Compared to (Albert et al., 2017), the OA is improved by 3.3% and 5.9% in Hameln and Schleswig, respectively. Our future work will focus on improving the classification of *small* polygons.

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.
The first author is an associate member of the Research Training Group i.c.sens (GRK 2159), funded by the German Research Foundation (DFG).

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., 2017. A higher order conditional random field model for simultaneous classification of land cover and land use. ISPRS Journal of Photogrammetry and Remote Sensing 130: 63-80.

Audebert, N., Saux, B. L., Lefevre, S., 2018. Beyond RGB: Very high resolution urban remote sensing with multimodal deep networks. ISPRS Journal of Photogrammetry and Remote Sensing 140: 20-32

Badrinarayanan, V., Kendall, A., Cipolla, R., 2017. SegNet: A deep convolutional encoder-decoder architecture for image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence 39(12): 2481-2495.

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

Gerke, M., 2015. Use of the stair vision library within the ISPRS 2d semantic labelling benchmark (Vaihingen). Tech. rep., International Institute for Geo-Information Science and Earth Observation.

He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition. IEEE Conference on Computer Vision and Pattern Recognition, pp. 770-778.

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.

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

Long, J., Shelhamer, E., Darrell, T., 2014. Fully convolutional networks for semantic segmentation. IEEE Conference on Computer Vision and Pattern Recognition, pp. 3431-3440.

Lin, G.S., Milan, A., Shen, C.H., Reid, I., 2017a. RefineNet: multi-path refinement networks for high-resolution semantic segmentation. IEEE Conference on Computer Vision and Pattern Recognition, pp. 1925-1934

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

Maggiori, E., Tarabalka, Y., Charpiat, G., Alliez, P., 2017. High-resolution semantic labelling with convolutional neural networks. IEEE Transactions on Geosciences and Remote Sensing, Vol. 55 (12), pp. 7092-7103

Marmanis, D., Schindler, K., Wegner, J. D., Galliani, S., Datcu, M., Stilla, U., 2018. Classification with an edge: Improving semantic image segmentation with boundary detection. ISPRS Journal of Photogrammetry and Remote Sensing 135: 158–172.

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 Photogrammetry, Remote Sensing and Spatial Information Sciences II-3/W4, pp. 143–148.

Noh, H., Hong, S., Han, B., 2015. Learning deconvolution network for semantic segmentation. IEEE International Conference on Computer Vision, pp. 1520-1528.

Ren S., He, K., Girshick, R., Sun, J. 2015. Faster R-CNN: towards real-time object detection with region proposal networks. Advances in Neural Information Processing Systems 28, pp. 91-99

Ronneberger O., Fischer P., Brox T., 2015. U-Net: Convolutional neworks for biomedical image segmentation. Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015, pp. 234-241

Volpi, M., Tuia, D., 2018. Deep multi-task learning for a geographically-regularized semantic segmentation of aerial images. ISPRS Journal of Photogrammetry and Remote Sensing 144: 48-60.

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.

Wegner, J.D., Rottensteiner, F., Gerke, M., Sohn, Gunho, 2017. The ISPRS labelling challenge. Available in the WWW: http://www2.isprs.org/commissions/comm3/wg4/semantic-labeling.html (accessed 09/01/2019).

Xie, S.N., Tu, Z.W., 2017. Holistically-Nested Edge Detection. International Journal of Computer Vision, Vol. 125 (3), pp. 3-18

Yang, C., Rottensteiner, F., Heipke, C., 2018: Classification of land cover and land use based on convolutional nerual networks. ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences. Vol. IV-3, pp. 251-258

Zhao, H.S., Shi, J.P., Qi, X.J., Wang, X.G., Jia, J.Y., 2017. Pyramid scene parsing network. IEEE Conference on Computer Vision and Pattern Recognition, pp. 2881-2890

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, Vol. 5, pp. 8-36.