IMPROVING THE CLASSIFICATION OF LAND USE OBJECTS USING DENSE CONNECTIVITY OF CONVOLUTIONAL NEURAL NETWORKS

A. Gujrathi[1,2] *, C. Yang[1], F. Rottensteiner[1], KM. Buddhiraju[2], C. Heipke[1]

[1] Institute of Photogrammetry and GeoInformation, Leibniz Universität Hannover, Germany - (gujrathi, yang, rottensteiner, heipke)@ipi.uni-hannover.de
[2] Centre of Studies in Resources Engineering, Indian Institute of Technology Bombay, India - bkmohan@csre.iitb.ac.in

Commission II, WG II/6

KEY WORDS: Land use classification, CNN, Geospatial land use database, DenseNet, global average pooling

ABSTRACT:

Land use is an important variable in remote sensing which describes the functions carried out on a piece of land in order to obtain benefits and is especially useful to the personnel working in the fields of urban management and planning. The land use information is maintained by national mapping agencies in geo-spatial databases. Commonly, land use data is stored in the form of polygon objects; the label of the object indicates land use. The main goal of classification of land use objects is to update an existing database in an automatic process. Recently, Convolutional Neural Networks (CNN) have been widely used to tackle this task utilizing high resolution aerial images (and derived data such as digital surface model). One big challenge classifying polygons is to deal with the large variation in their geometrical extent. For this challenge, we adopt the method of Yang et al. (2019) to decompose polygons into regular patches of fixed size. The decomposition leads to two sets of polygons: *small* and *large*, where the former suffers from a lower identification rate. In this paper, we propose CNN methods which incorporate *dense connectivity* and integrate it with intermediate information via *global average pooling* to improve land use classification, mainly focusing on *small* polygons. We present different network variants by incorporating intermediate information via *global average pooling* from different stages of the network. We test our methods on two sites; our experiments show that the dense connectivity and integration of intermediate information has a positive effect not only on the classification accuracy on the whole but also on the identification of small polygons.

1. INTRODUCTION

Land use is an important variable in remote sensing which describes the socio-economic function of a piece of land in order to obtain benefits (Barnsley & Barr 2000). In the region of central Europe, the government surveying authorities maintain geospatial database containing objects whose boundaries are related to property boundaries. The information of land use of property objects becomes outdated quickly as the property owners are not obliged to inform the government of changes in land use. Thus, a system is required to analyse the change in land use of the objects stored in the geospatial database. This can be done by extracting land use information from recently acquired aerial images. The extracted information is checked against the information stored in the database and thus a database update can be performed (Gerke & Heipke, 2008; Albert et al., 2017).

The land use information is maintained by national mapping agencies in geo-spatial databases in the form of polygon objects with class labels indicating the object's land use. This setting is adopted in this paper, where the primitive considered for land use classification is a polygon object of the geospatial database. The main goal of land use classification is to update an existing database in an automatic process. Traditional approaches for land use classification require hand-crafted features derived from image data, and then apply a supervised classifier such as Random Forests to deal with these features. Here, contextual models like Conditional Random Fields (CRF) have also been applied for classification purpose, e.g. (Albert, et al., 2017). However, these methods incorporating hand-crafted features are strenuous and time consuming. The rapid progress in remote sensing technology has resulted in a bulk of images of the earth surface taken by satellites, airplanes or drones, with different imaging modalities. With the large availability of data, the focus shifts to the automatic extraction of valuable information. Approaches based on CNN are known to provide impressive results when large amount of training data is available; CNNs are currently being used in many remote sensing applications (Zhu et al., 2017).

For land use classification, a major challenge is the large variation of polygons in terms of their geometrical extent; for instance, *road* objects are thin and long, whereas *residential* objects cover both, very large and quite small areas. Recently, CNN-based method for land use classification proposed by Yang et al. (2019) solved the problem by decomposing large polygons into smaller patches of fixed size which suits the input of CNN. To represent a polygon, they use a combination of its shape in the form of a binary mask and the image data (e.g. RGB) and decompose it to form patches of fixed size. We adopt this methodology for the generation of input patches from polygons. During the decomposition, two types of polygons are differentiated: *large* polygons have multiple smaller patches whereas *small* polygons have exactly one patch. In the analysis of their classification results, the authors observed that the *small* polygons are hard to be classified correctly. Possible reasons for lower classification accuracy of *small* polygons are the following: (i) One problem of CNN is that as input passes through many layers of a neural network, the information can vanish by the time it reaches the end of the network. (ii) The final 1-D feature vector before classification may not capture valid information of the *small* polygons due to many pooling operations.

* Corresponding author.

In this paper, we build on the methods proposed in (Yang et al., 2019) with the aim of improving the classification of land use objects, mainly focussing on *small* polygons. In our work, we only use the binary mask and RGB data as input. The scientific contribution of this paper can be summarized as follows:

- We propose a network architecture incorporating dense connectivity (Huang et al., 2017) that strengthens information flow to improve the land use classification. The key is to create short paths from early layers to later layers, maximizing the data flow through the network.
- We apply global average pooling (GAP) (Lin et al., 2013) at different stages of the network, resulting in many network variants, and utilize it as intermediate information in the classification process, to compensate the data loss caused by the many pooling operations in the network.
- We conduct an extensive set of experiments to compare these network variants, and to highlight the benefits and drawbacks of the proposed methods.

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

2. RELATED WORK

We start with a brief introduction to land use classification. We then briefly discuss a history of deep learning (especially CNN) in land use classification. After that, we present the current state-of-the-art in land use classification based on polygon objects in geospatial databases.

Land cover is the physical material present on a piece of land (e.g., *water*, *grass*, *concrete* etc.). Land use corresponds to the socio-economic function of a piece of land (e.g., *residential*, *agricultural* etc.). Classification of land cover is simpler because there is a direct relationship between land cover and exitant spectral reflectance, but land use is an abstract concept. The technique suggested in Barnsley & Barr (2000) for land use classification is to divide the classification procedure into two stages: the first being semantic segmentation of the image for land cover classification; the second being land use classification based on the spatial pattern of land cover. The first stage can be performed by a number of techniques ranging from a standard maximum likelihood classifier to artificial neural networks. The disadvantage of such a two stage process is that the accuracy of the land use classification depends on the accuracy of land cover classification, i.e., an error in the first stage is propagated through the second stage. Johnsson (1994) and Bauer & Steinnocher (2001) investigated segment-based land use classification. Segments are obtained by spectral classification. Spatial information of segments such as size, neighbours etc., are used for rule-based classification of image segments into land use categories. An interesting work on land use object classification combining high spatial resolution imagery, LiDAR data and cadastral plots in given in Hermosilla et al. (2012). Land use objects are characterised by image based, geometric and contextual hand crafted features. With the emergence of classifiers that work on both spatial and spectral dimensions, e.g., neural network classifier, it is possible to perform land use classification is one step.

As computers became more powerful and processing speed increased, computationally intensive but flexible neural network based classification has become more attractive. The LeNet-5 architecture (LeCun et al., 1998) is one of the first successful applications of CNN and is the origin of most of the recent architectures. The building blocks of LeNet-5 are convolution, pooling and non-linearity layers. Then, Alexnet (Krizhevsky et al., 2012), a deep neural network architecture provided a seismic shift in the field of image classification. Another variant of classifiers called Support Vector Machines (SVMs) are frequently used for solving image classification problems. SVMs are independent of the dimensionality of feature space, therefore provide better classification results with limited training samples. Neural networks and SVMs show comparable results for land use classification (Dixon et al., 2008). However, neural network based classification is more robust to training site heterogeneity; and such heterogeneity is common in remote sensing images (Paola & Schowengerdt 1995).

As mentioned in Section 1, the first challenge in the classification of land use polygons using CNN is the variation in geometric extent of polygons. To the best of our knowledge, LiteNet (Yang et al., 2018) is the first architecture to perform classification of land use polygons using CNN. The network was trained separately using RGB data and a label image encoding land cover. The input patches for CNN were generated by decomposing the polygons. In the input patch, the area inside the polygon is represented by RGB data or land cover encoding and the area outside the polygon is set to 0. However, this underutilization of data leads to a loss of context information. Yang et al. (2019) represent a polygon using a combination of its shape in the form of a binary mask and the image data (e.g. RGB), finally decomposing it to form patches of a fixed size. We adopt this methodology for patch generation from polygons. LuNet (Yang et al., 2019), which is based on LiteNet, consists of four convolutional blocks and two branches towards the end called two-branch-convolution. The upper branch of the two-branch-convolution extracts global features that are representative of the complete image. The lower branch uses a region of interest (ROI) to focus on the most relevant regions in the image, which helps in the classification of polygons. We also adopt this two-branch convolution in our architecture, as it was demonstrated to enhance the classification of land use polygons.

Another work on urban land use classification using object based CNN is presented in Zhang et al. (2018). The objects generated using mean shift clustering algorithm are classified into two types: linearly and non-linearly shaped objects. Two CNNs with different model structures and window sizes predict the labels for linearly and non-linearly shaped objects and a rule based decision fusion is performed to combine the results. However, such two-scale feature representation might be insufficient to characterize complex geometric polygons. A joint deep learning framework for land cover and land use classification that involves Multi Layer Perceptron (MLP) and CNN classification models was proposed in Zhang et al. (2019). The intrinsically hierarchical relationships between land cover and land use were modelled via an iterative Markov process. However, their method focuses solely on urban and suburban areas, leading to an insufficient model transferability.

Recent work by He et al. (2016) and Huang et al. (2017) has shown that shorter connections between layers close to input and those close to output in very deep CNNs leads to more accurate and efficient to train networks; ResNet (He et al., 2016) uses identity connections to bypass signal and summation operations when combining input and output layers. These networks are easier to optimize and gain accuracy from considerably increased depth. Many ResNet layers contribute very little and there is a large amount of redundancy in deep residual networks. Stochastic depth (Huang et al., 2016) randomly drops the layers

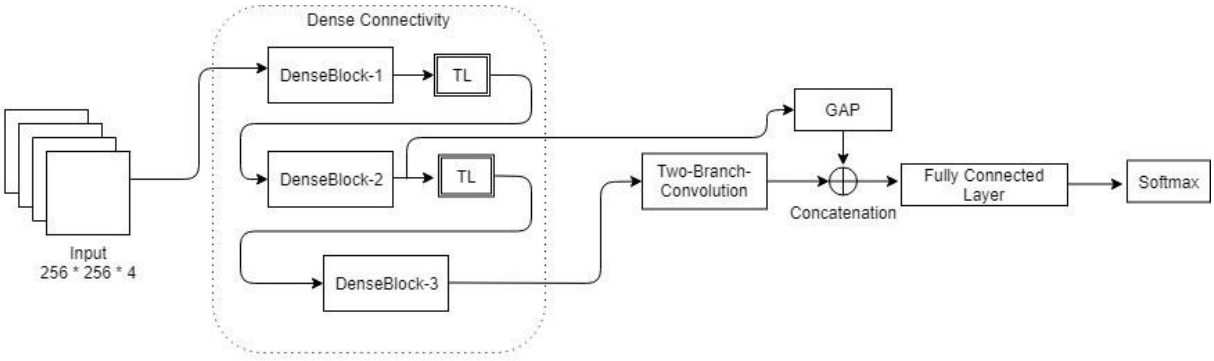

Figure 1. The architecture of DenseLuNet-2. TL: Transition layer, DenseBlock: cf. Fig. 2, Two-Branch-Convolution: cf. Yang et al. (2019)

during training to overcome this problem. Feed-forward neural network can be considered as an algorithm with a state variable, where the state is passed on from layer to layer. Every neural network layer reads the state from its previous layer and writes to the subsequent layer its own state in addition to the previous state. The network architectures that make the state preservation implicit are desirable to overcome redundancy in network layers.

The DenseNet architecture (Huang et al., 2017) differentiates between the information that is added to the network and information that is preserved. DenseNet allows maximum information flow within the network, by connecting all layers within a dense block. The DenseNet architecture encourages improved flow of information and gradients throughout the network, alleviates the vanishing-gradient problem, and helps in strengthening feature propagation. Also, this architecture significantly reduces the number of parameters to be learnt and encourages feature reuse. GAP (Lin et al., 2013) computes the average value of each feature map at a particular layer of the network. An advantage of GAP is that it sums up the spatial information which might be useful in classification of data. GAP also introduces global context (Yu et al., 2018) providing high level sematic information.

Our approach follows the concepts of Huang et al. (2017) and Lin et al. (2013). We use dense block as main classification unit and GAP to obtain intermediate information, which we believe helps in feature propagation and compensates the data loss in our CNN architecture.

3. LAND USE CLASSIFICATION USING CNN

In this section, we propose a CNN for land use classification which is based on LuNet (Yang et al., 2019). As mentioned earlier, the large variation of polygons in terms of geometrical extent is a challenge, because our CNN requires a fixed input size (256 x 256 pixels) while returning a land use label. In this work, the way in which the image patches are prepared follows the method of Yang et al., (2019), which is introduced in section 3.1. The concept of dense connectivity is introduced in section 3.2. Section 3.3 outlines the network architecture used for land use classification. Section 3.4 describes the network variants and section 3.5 describes the procedure.

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 (RGB bands and binary object mask) 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 *cropping*: 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. Finally, the results for the individual input patches are combined (cf. section 3.5).

3.2 Dense connectivity

We adopt the *dense block* concept from Huang et al. (2017) as network component for classification. The key is to create short paths from early layers to later layers, maximizing the data flow through the network. The spatial size of feature maps remains constant in a dense block (Fig. 2), where each layer within the block obtains input (i.e. feature maps) from all the previous layers of the block. Suppose, each layer in a dense block produces k feature maps, then the l^{th} layer has $n + k \times (l - 1)$ input feature maps, where n is the number of input feature maps to the dense block. The feature maps from previous layers of the dense block are concatenated to build the feature maps of the l^{th} layer. The number of feature maps generated by each layer within a dense block, k, is called growth rate (Huang et al. 2017), which is very small ($k = 12$ in our paper), thus adding only a small number of feature maps at every layer. Therefore, if there are L layers in a dense block, there are $\big(L \times (L + 1)\big) \div 2$ connections, as opposed to just L connections in a traditional CNN architecture (Krizhevsky et al., 2012).

A dense block can consist of an arbitrary number of layers (we use 4 layers per dense block in our paper). Each layer in the dense block performs a composite function of three consecutive operations: batch normalization (BN), rectified linear unit (ReLU) processing and 3×3 convolution (Conv). According to Huang et al. (2017), the dense connectivity strengthens feature propagation which is the key of its success in visual recognition.

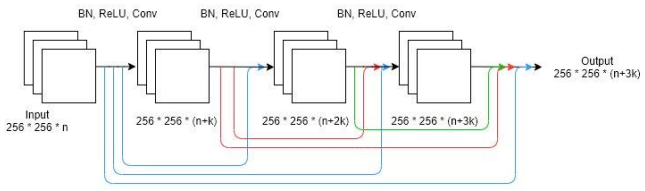

Figure 2. A 3-layer dense block with n input channels and k growth rate. Please refer to texts for the abbreviations.

Network Variant	F1 [%]										avg. F1 [%]	OA [%]
	res.	*non-res.*	*green*	*traf.*	*square*	*cropl.*	*grassl.*	*forest*	*water*	*others*		
Hameln												
LuNet	76.5	60.5	57.1	87.8	40.2	55.9	30.9	66.0	34.6	46.0	55.6	69.2
DenseLuNet	81.2	65.8	70.8	89.5	**47.9**	**73.1**	**32.6**	66.9	34.3	43.7	60.6	74.0
DenseLuNet-1	84.4	69.4	**74.8**	87.9	44.8	72.8	26.3	**72.8**	**38.9**	44.0	**61.6**	74.9
DenseLuNet-2	82.6	67.4	71.0	**89.6**	41.0	67.1	20.5	68.2	37.5	**48.6**	59.4	74.4
DenseLuNet-12	**84.8**	**69.6**	72.7	**89.6**	39.6	70.2	25.9	70.7	35.8	47.6	60.7	**75.8**
Schleswig												
LuNet	79.4	27.1	58.2	87.7	14.9	73.5	76.5	78.1	57.1	28.1	58.1	70.6
DenseLuNet	77.6	51.9	48.3	**88.1**	12.7	71.2	73.7	79.3	53.3	26.1	58.2	69.8
DenseLuNet-1	80.7	56.3	59.3	86.8	19.2	76.1	78.9	77.9	**61.6**	25.8	62.2	72.9
DenseLuNet-2	82.9	57.2	60.1	87.8	**22.7**	75.6	75.5	**81.3**	58.6	**30.7**	**63.2**	73.4
DenseLuNet-12	**84.6**	**58.1**	**61.8**	87.8	20.4	**77.0**	**79.0**	76.0	53.5	20.8	61.9	**74.5**

Table 1. Results of land use classification. Network variants cf. section (3.4). F1: F1 score, OA: Overall Accuracy, both evaluated on the basis of objects. Best scores are printed in bold.

3.3 DenseLuNet

This network is based on LuNet (Yang et al., 2019) and consists of three dense blocks (cf. Section 3.2) with transition layers between them. A *transition layer* (TL) consist of BN, ReLU, 3×3 convolution and 2×2 max-pooling with stride 2 and the number of output channels is equal to the number of input channels. TL facilitates down-sampling in our network. Every dense block contains four layers, each layer generates 12 feature maps. After the last dense block, two-branch convolution (Yang et al., 2019) is applied for generating a 512 dimensional feature vector for classification. The upper branch of the two-branch-convolution extracts global features that are representative of the complete image by performing max-pooling, followed by three convolution layers, BN and ReLU. The lower branch uses an ROI, to focus on the most relevant regions in the image. In this branch, we focus on these regions by aligning a rectangular image grid enclosing the polygon. The output of the two branches are concatenated and given as input to the fully connected layer. The fully connected layer delivers a vector of class scores $(Z_{LU^1}, \dots, Z_{LU^M})^T$, where $\mathbb{C}_{LU} = \{C_{LU^1}, \dots, C_{LU^M}\}$ 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 obtain a probabilistic class score, the softmax function is applied to the class scores:

$$P(C_{LU^c}|X) = softmax(Z_{LU}, C_{LU^c}) = \frac{exp(Z_{LU})}{\sum_{i=1}^{M} exp(Z_{LU^i})}, \qquad (1)$$

Training is based on mini-batch Stochastic Gradient Descent (SGD) and step learning policy. The function to be optimized is the cross-entropy loss:

$$L = -\frac{1}{N} \cdot \sum_{c,k} \left[y_{LU^c}^k \cdot \log\left(P(C_{LU^c}|X_k)\right) \right], \qquad (2)$$

where X_k is the k^{th} image in the mini-batch, N is the number of images in a mini-batch, $y_{LU^c}^k$ is 1 if the training label of X_k is C_{LU^c} and 0 otherwise.

3.4 Network variants

The many stages of convolution and pooling operations can cause the final 1-D feature vector to capture no valid information of the input image. The intermediate information from different pooling stages could be helpful for classification. We introduce the intermediate information via GAP (Lin et al., 2013). GAP, when applied on the output of a network layer, computes the average value of each feature map and results in a 1-D vector. GAP is performed on the output of *dense block* and is concatenated to the 1-D feature vector obtained from the two-branch convolution (Yang et al., 2019), which serves as the final feature vector for classification.

In this paper, we investigate four network variants differing by the stages at which the intermediate information using GAP is extracted on the DenseLuNet base architecture: i). DenseLuNet architecture as described in Section 3.3. ii). Applying the GAP at the output of the first dense block of DenseLuNet, referred to as DenseLuNet-1. iii). Applying the GAP at the output of the second dense block of DenseLuNet, referred to as DenseLuNet-2 (cf. Fig. 1). iv). Applying the GAP at the output of the first and second dense block of DenseLuNet, referred to as DenseLuNet-12. For training these variants, the mini-batch size is set to 10. All networks are trained for five epochs, using a base learning rate of 0.001 and reducing it to 0.0001 after two epochs.

3.5 Inference of polygons

All network variants output a probabilistic score for each patch. If a polygon results in exactly one patch during cropping, its prediction is straightforward, the prediction score of the polygon is the same as the patch score; if a polygon is split into multiple patches, the product of the probabilistic patch scores is determined first, and then the prediction is made based on this product.

4. EXPERIMENTS

4.1 Datasets and test setup

4.1.1. Datasets: Our experiments for classification of land use are evaluated on two test sites, located in the cities of Hameln and Schleswig (Germany). Hameln covers an area of $2\ km \times 6\ km$. It contains densely built-up residential areas in the centre of the city as well as detached houses, rural areas, industrial areas and rivers. Schleswig covers an area of $6\ km \times 6\ km$, showing similar characteristics as Hameln. For both Hameln and Schleswig, digital orthophotos (DOP), and land use objects (corresponding to cadastral parcels) from the German Authoritative Real Estate Cadastre Information System (ALKIS) are available. The DOP are multispectral images (RGB + infrared / IR) with a ground sampling distance (GSD) of $20\ cm$. The reference for land use is derived from the German geospatial land use database.

We distinguish 10 land use classes for the Hameln and Schleswig test sites: *residential (res.), non-residential (non-res.), urban green (green), traffic (traf.), square, cropland (cropl.), grassland (grassl.), forest, water body (water)* and *others*. The class structure of land use is same as in (Yang et al., 2019).

4.1.2. Test setup: There are 2945 polygons in Hameln and 4345 polygons in Schleswig. Each test data set is split into two blocks for cross validation. The block size is 10000×15000 pixels ($6\ km^2$) and 30000×15000 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. We evaluate land use classification based on the number of correctly classified database objects. We report overall accuracy (OA), i.e., the percentage of land use objects assigned the correct class label by the classification process, and F1 score, i.e., the harmonic mean of precision and recall. All the networks were implemented using tensorflow framework (Abadi et al., 2015). We use a GPU (Nvidia GeForce GTX 1080 TI, 11GB) to accelerate training and inference.

We perform data augmentation on the patches generated from cropping. Here, we differentiate two scenarios: *Large polygons*, i.e. polygons that had to be split because they do not fit into the input window of the CNN, are augmented by horizontal and vertical flipping and by applying random rotations in intervals of 30°. In the other case, i.e. *small polygons* which fit the input size of the CNN, are augmented by horizontal and vertical flipping and by applying random rotations in intervals of 5°. In the end, there are 354178 and 479978 patches for Hameln and Schleswig, respectively.

4.2 Evaluation of land use classification

4.2.1. Evaluation and comparison of network variants: In this section, we compare four variants of networks (cf. Section 3.4) using two datasets Hameln and Schleswig. The LuNet network serves as a baseline for all other variants. The evaluation results for land use classification evaluated on land use objects are given in Table 1. The best values achieved for every accuracy measure on each dataset are printed in bold font. To summarize the performance of the models, the F1 scores with respect to each land use class along with average F1 scores and OA are provided. Analysing Table 1, it is evident that DenseLuNet and its variants perform better than LuNet in terms of either OA or average F1 score on both datasets. The best performing variant on Hameln is DenseLuNet-12 which shows an improvement of 6.6% and 5.1% in OA and F1 scores, respectively, in comparison with LuNet. For Schleswig, an improvement of 3.9% and 3.8% in terms of OA and F1 scores, respectively, was reached by DenseLuNet-12, which is the best performing model on this dataset, in comparison with LuNet. On the contrary, DenseLuNet shows about 1% decrease in OA on Schleswig dataset, whereas the F1 score remains the same. The reason for this is unclear and requires further investigation. Overall, we point out that incorporating dense connectivity leads to better classification results.

In general, all the network variants face difficulties in classifying objects belonging to the classes *square*, *grassland* and *others* which can be attributed to the fact that only a very small amount of training data is available for these classes, also *others* is a class of heterogeneous appearance. DenseLuNet-1 shows highest improvement of the F1 score for the class *green* by a margin of 17.7% on Hameln. On Schleswig, DenseLuNet-12 shows the highest improvement by a margin of 31% on *non-residential*.

4.2.2. Effectiveness of using global average pooling: In our network variants DenseLuNet-1 and -2, we apply GAP at the output of the 1st and 2nd dense block, respectively, and concatenate it to the 1-D feature vector obtained towards the end of the network. In the variant DenseLuNet-12, we apply GAP at the output of both 1st and 2nd dense block. We believe that the intermediate information computed using GAP is helpful in the classification as it can compensate for information that was lost due to many pooling operations in the network. Analysing Table 1, it is easy to notice that the DenseLuNet variants with GAP perform better than DenseLuNet on both, Hameln and Schleswig in terms of either OA or average F1 score. However, when compared to the performance of DenseLuNet, for Hameln the difference is not so pronounced: DenseLuNet-2 shows a slight decrease (1.2%) in average F1 score and OA being almost identical when compared to DenseLuNet. However, on the Schleswig dataset, improvements are seen in both OA and average F1 score by all the three DenseLuNet variants incorporating GAP. Therefore, we consider that GAP has a positive impact on the classification of land use polygons.

Among the three DenseLuNet variants incorporating GAP, DenseLuNet-12 is the best performing variant on both Hameln and Schleswig in terms of OA, although, the results pertaining to average F1 score do not show a particular trend. We take this as an indication that the more intermediate information added to the classification process, the better are the classification results.

4.2.3. Influence of the object size: Table 2 shows the OA and average F1 scores of *small* and *large* polygons along with the combined results which are the same as the ones shown in Table 1. The results are given for all the network variants on Hameln and Schleswig. The *small* set consists of polygons that were represented as a single patch in the classification process. The

Network Variant	Hameln						Schleswig					
	OA[%]			avg. F1 [%]			OA[%]			avg. F1 [%]		
	Large (1955)	*Small (990)*	*All (2945)*	*Large (1955)*	*Small (990)*	*All (2945)*	*Large (3435)*	*Small (910)*	*All (4345)*	*Large (3435)*	*Small (910)*	*All (4345)*
LuNet	72.5	62.7	69.2	56.5	38.9	55.6	73.9	58.1	70.6	58.5	39.7	58.1
DenseLuNet	76.7	68.7	74.0	60.5	47.6	60.6	74.1	53.7	69.8	57.5	39.1	58.2
DenseLuNet-1	78.2	68.4	74.9	**63.0**	45.9	**61.6**	77.1	57.1	72.9	62.4	**43.6**	62.2
DenseLuNet-2	77.6	68.1	74.4	59.9	**49.5**	59.4	77.5	57.7	73.4	**63.6**	42.2	**63.2**
DenseLuNet-12	**79.1**	**69.2**	**75.8**	62.3	48.2	60.7	**78.4**	**59.7**	**74.5**	62.0	42.0	61.9

Table 2. Results of land use classification represented separately for large, small and all polygons (cf. Table 1). The results are provided for all the network variants on Hameln and Schleswig dataset. The number of polygons in each set is given in parenthesis.

large set consists of polygons that were split into patches during the input patch generation (cf. Section 4.1.2). In general, the large set accuracy is greater when compared to the small set because large numbers of patches belonging to the large set are available during classification. DenseLuNet-12 shows best performance on Hameln and Schleswig in terms of OA of *small*, *large* and *all* polygons, however, the average F1 scores do not show a particular trend. Coming to the classification of *small* set, DenseLuNet-12 shows 6.5% improvement in the OA in comparison to LuNet on Hameln. This can be attributed to a maximization of data flow due to dense connectivity and utilization of intermediate information from two stages of the network. However, in Schleswig, DenseLuNet-12 shows 1.6% improvement in the OA of *small* set in comparison to LuNet, while the other DenseLuNet variants show similar performance to that of LuNet in classification of *small* polygons.

5. CONCLUSION

In this paper, we proposed a CNN architecture for classification of land use objects in a geospatial database incorporating dense connectivity; we call it DenseLuNet. We investigate four variants of networks differing by the stages at which the intermediate information is extracted using GAP on two test sites. DenseLuNet and its variants perform better than LuNet (Yang et al., 2019) in terms of either overall accuracy or the average F1 score on both datasets. Also, we observe that intermediate information obtained using GAP has a positive impact on the classification of land use polygons. Compared to LuNet, DenseLunet-12 shows an improvement of 6.6% and 5.1% in OA and F1 scores, respectively, for the Hameln dataset. DenseLuNet-12 shows best performance on Hameln and Schleswig in terms of OA of *small*, *large* and *all* polygons. We conclude that the more the intermediate information via GAP is utilized in the classification process, the better are the classification results.

Future research should focus on including more object knowledge, e.g., in terms of height information. We are also interested to incorporate a hierarchical and more detailed class structure (Yang et al., 2020) into our approach and to investigate the influence of partly incorrect training data; the latter as a way to be able to use large parts of existing geospatial database content for training. Although some of that information will be outdated and thus wrong, the problem of needing vast amounts of training data could be alleviated in this way. Finally, dense connectivity requires significant amounts of GPU memory and we faced memory issues implementing the network with more than three dense blocks. To overcome these issues, network implementations using shared memories and gradient checkpointing (Pleiss et al., 2017) can be performed.

ACKNOWLEDGEMENTS

We thank the Landesamt für Geoinformation und Landesvermessung Niedersachsen(LGLN), the Landesamt für Vermessung und Geoinformation Schleswig Holstein (LVermGeo) and the Landesamt für innere Verwaltung Mecklenburg-Vorpommern (LaiV-MV) for providing the test data and for their support of this project. The first author is a Master's student at Centre of Studies in Resources Engineering, Indian Institute of Technology Bombay and a Combined Study and Practice Stays for Engineers from Developing Countries (KOSPIE) Scholar, funded by Deutscher Akademischer Austauschdienst (DAAD), whose support is gratefully acknowledged.

REFERENCES

Abadi, M. et al. (2015). Large-scale machine learning on heterogeneous systems. https://www.tensorflow.org (accessed 11/04/2020).

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.

Bauer, T., & Steinnocher, K. (2001). Per-parcel land use classification in urban areas applying a rule-based technique. GeoBIT/GIS, 6, 24-27.

Barnsley, M. J. & Barr, S. L. (2000). Monitoring urban land use by earth observation. Surveys in Geophysics 21(2): 269-289.

Dixon, B., & Candade, N. (2008). Multispectral landuse classification using neural networks and support vector machines: one or the other, or both? Int. J. RS, 29(4), 1185-1206.

Gerke, M. & Heipke, C. (2008). Image based quality assessment of road databases. Int. J. of Geoinf. Science, 22 (8), 871-894.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. CVPR, 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.

Huang, G., Sun, Y., Liu, Z., Sedra, D., & Weinberger, K. Q. (2016). Deep networks with stochastic depth. ECCV, 646-661.

Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely connected convolutional networks. CVPR, 4700-4708.

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

Johnsson, K. (1994). Segment-based land-use classification from SPOT satellite data. PE&RS, 60(1), 47-54.

Krizhevsky, A., Sutskever, I. & Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. NIPS'12, 25 Vol. 1, 1097-1105.

LeCun, Y., Bottou, L., Bengio, Y. & Haffner, P. (1998). Gradient-based learning applied to document recognition. Proceedings of the IEEE 86(11): 2278–2324.

Lin, M., Chen, Q., & Yan, S. (2013). Network in network. arXiv preprint arXiv:1312.4400.

Paola, J. D., & Schowengerdt, R. A. (1995). A detailed comparison of backpropagation neural network and maximum-likelihood classifiers for urban land use classification. IEEE Transactions on Geoscience and remote sensing, 33(4), 981-996.

Pleiss, G., Chen, D., Huang, G., Li, T., van der Maaten, L., & Weinberger, K. Q. (2017). Memory-efficient implementation of densenets. arXiv preprint arXiv:1707.06990.

Yang, C., Rottensteiner, F., & Heipke, C. (2018). Classification of land cover and land use based on convolutional neural networks. ISPRS Annals IV-3, 251-258.

Yang, C., Rottensteiner, F., & Heipke, C. (2019). Towards better classification of land cover and land use based on convolutional neural networks. International Archives XLII-2/W13, 139-146.

Yang, C., Rottensteiner, F., & Heipke, C. (2020). Exploring semantic relationships for hierarchical land use classification based on convolutional neural networks. ISPRS Annals, V-B2.

Yu, C., Wang, J., Peng, C., Gao, C., Yu, G., & Sang, N. (2018). Learning a discriminative feature network for semantic segmentation. CVPR, 1857-1866.

Zhang, C., Sargent, I., Pan, X., Li, H., Gardiner, A., Hare, J., & Atkinson, P. M. (2018). An object-based convolutional neural network (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.