Semantic Segmentation of Small Objects and Modeling of Uncertainty in Urban Remote Sensing Images Using Deep Convolutional Neural Networks

Michael Kampffmeyer*, Arnt-Børre Salberg† and Robert Jenssen*

*Machine Learning @ UiT Lab, UiT–The Arctic University of Norway
†Norwegian Computing Center

Abstract

We propose a deep Convolutional Neural Network (CNN) for land cover mapping in remote sensing images, with a focus on urban areas. In remote sensing, class imbalance represents often a problem for tasks like land cover mapping, as small objects get less prioritised in an effort to achieve the best overall accuracy. We propose a novel approach to achieve high overall accuracy, while still achieving good accuracy for small objects. Quantifying the uncertainty on a pixel scale is another challenge in remote sensing, especially when using CNNs. In this paper we use recent advances in measuring uncertainty for CNNs and evaluate their quality both qualitatively and quantitatively in a remote sensing context. We demonstrate our ideas on different deep architectures including patch-based and so-called pixel-to-pixel approaches, as well as their combination, by classifying each pixel in a set of aerial images covering Vaihingen, Germany. The results show that we obtain an overall classification accuracy of 87%. The corresponding F1-score for the small object class "car" is 80.6%, which is higher than state-of-the art for this dataset.

1. Introduction

Object detection, mapping of land cover and change detection have historically been some of the the most important tasks in remote sensing and find application in, among others, environmental monitoring, agriculture, forestry, and urban planning. For instance, a high quality and updated land cover map is required by local government agencies that are interested in large-scale analysis to automatically extract useful geographical features, by economic forecasters that are interested in how much business a particular retail store did conduct by counting cars in the parking lot, or by relief agencies that are interested in knowing the hardest hit areas after a natural disaster.

Remote sensing imagery is often characterized by complex data properties in the form of heterogeneity and class imbalance, as well as overlapping class-conditional distributions [6]. Together, these aspects constitute severe challenges for creating land cover maps or detecting and localizing objects, producing a high degree of uncertainty in obtained results, even for the best performing models [21, 24].

In recent years, deep CNNs have emerged as the leading modeling tools for image pixel classification and segmentation in general [14, 22], and have had an increasing impact also in remote sensing [21, 24, 25, 28]. This increasing interest is reflected for example in the ISPRS semantic segmentation challenge [1], where deep CNNs are dominating and are shown to provide the best performing models.

In this paper we apply and develop two recent CNN architectures, patch-based and pixel-to-pixel based, for segmentation of urban remote sensing images to map land cover with a focus on small objects. We study the potential of the cross-entropy loss function weighted with median frequency balancing, a loss function that was proposed by Eigen and Fergus [9] in the last few months, to improve segmentation accuracy for small classes in urban remote sensing, which to the authors knowledge has not been done previously. Inspired by the success of ensemble methods in the ImageNet competition [15, 32], we propose a combination of our models to achieve good overall classification performance, maintaining at the same time a good accuracy for small classes.

Of interest for this paper, is the recently introduced technique for quantifying uncertainty in deep learning by Gal and Ghahramani [12]. They showed that dropout training [31] in neural networks can be cast as approximate Bayesian inference in Gaussian processes. This means that uncertainty information can be extracted from models without requiring additional parameters. Performing dropout during the test phase can be viewed as performing Monte Carlo sampling from the posterior distribution over the var-

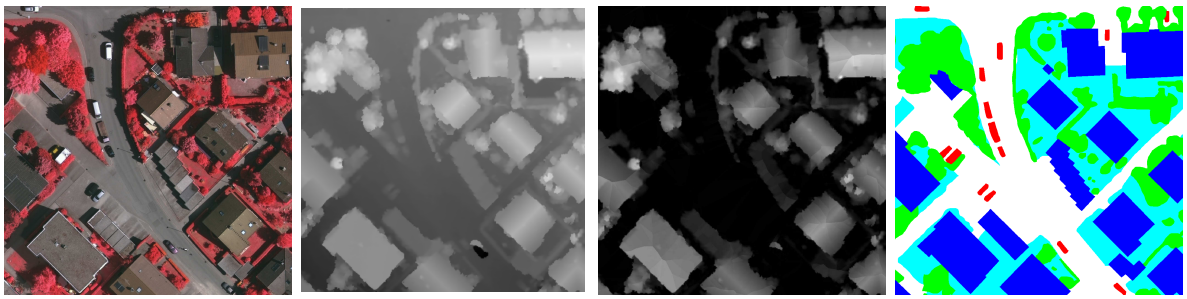

Figure 1: A small example patch from the validation dataset. From left to right: RGB image, DSM, normalized DSM, and ground truth image. It illustrates the difference in size between classes, such as the car class (red) and the building class (blue).

ious dropout models. This technique has been utilized in order to produce uncertainty maps as a visual aid e.g. in the context of segmentation of indoor scenes and outdoor camera videos [19]. To the authors knowledge, CNN model uncertainty in segmentation has previously only been used for visualization. One of our contributions in this work is a novel analysis of uncertainty maps that links uncertainty information and segmentation accuracy.

2. Related work

Our approach to segmentation builds on the recent successes that deep learning techniques have achieved for image segmentation. CNNs have been extensively used for the task of image classification [20], the task of localization [29] and the more challenging task of object detection both using bounding box [13, 27] and sliding window approaches [29].

Lately, CNNs have also been applied to the task of image segmentation. In practice, there are currently two main approaches to performing image segmentation using CNNs. The first one, which we refer to as patch-based, relies on predicting every pixel in the image by looking at the enclosing region of the pixel. This is commonly done by training a classifier on small image patches and then either classifying all pixels using a sliding window approach, or more efficiently, convert the fully connected layers to convolutional layers, thereby avoiding overlapping computations as described in Sermanet *et al*. [29]. Further improvements can be achieved using multi-scale approaches or by iteratively improving the results in a recurrent CNN [11, 26].

The second approach is based on the idea of pixel-to-pixel semantic segmentation using end-to-end learning [22]. It uses the idea of a fully convolutional network (FCN), consisting of an encoder and a decoder. The encoder is responsible for mapping the image to a low resolution representation, whereas the decoder provides a mapping from the low resolution representation to the pixel-wise predictions. Up-sampling is achieved using fractional-strided convolutions [22]. This approach has recently improved the state-of-the art performance on many image tasks and, due to the lack of fully-connected layers, allows pixel-wise predictions for arbitrary image sizes.

Previous work that has been published on the ISPRS challenge includes among others Paisitkriangkrai *et al*. [24], who proposed a scheme for semantic segmentation using a combination of a patch-based CNN and a random forest classifier that is trained on hand-crafted features. To increase the classification accuracy further, a conditional random field (CRF) was used to smooth the final pixel labeling results. Recently, a dense FCN approach has also been employed, however, the paper is not published yet. Besides CNNs, also graph based approaches have been previously tested on the dataset [23].

Other related approaches applied pre-trained CNNs and a sliding window approach to perform a pixel classification in remote sensing images [21, 25]. Additionally, a region-based approach combined with a pre-trained CNN was adopted to detect small objects in areal images [28].

Recently, focus has also been put on using geospatial data from e.g. geographical information system (GIS) databases or crowd-sourced cartographic maps to improve object detection and semantic segmentation performance [2, 3, 33] or to perform cross-view matching between street-level images and GIS maps [7].

3. Dataset

The remote sensing dataset used to evaluate our proposed method is the ISPRS Vaihingen 2D semantic labeling contest dataset [1]. The dataset consists of 33 images of varying size, ranging from approximately 3 million to 10 million pixels each, each one being an image patch of a high resolution true ortho photo (TOP) that was taken of Vaihingen, a small town in Germany, with a ground sampling distance of 9 cm. Besides the TOP images, the dataset contains the Digital Surface Model (DSM) for each of the 33 images with the same spatial resolution. Additionally, normalized

DSMs were provided by Gerke [23], to limit the effects of varying ground height. Ground truth images are available for 16 of the 33 images in which all pixels are labeled by one of 6 classes, namely Impervious surfaces, Building, Low vegetation, Tree, Car, Clutter/background. An example patch of a TOP image from the validation dataset is displayed in Figure 1 alongside the corresponding DSM, normalized DSM, and ground truth image.

The evaluation procedure defined by the ISPRS [1] was used to evaluate our results. The performance on the classes is measured by using the F1-score (the F1-score is defined as $2 \cdot \text{precision} \cdot \text{recall}/(\text{precision} + \text{recall})$) and the overall accuracy is measured as the percentage of pixels that were labeled correctly. To reduce the effect of class boundaries, the class boundaries were eroded with a disk of radius 3 and ignored during evaluation, as specified by the ISPRS.

To evaluate our method, the labeled part of the dataset is divided into a training and validation set. Following the example of Paisitkriangkrai *et al*. [24], the training set consists of 11 images (areas: 1, 3, 5, 7, 13, 17, 21, 23, 26, 32, 37) and the validation set of 5 images (areas: 11, 15, 28, 30, 34). There are some ambiguities in the dataset, where pixels are mislabeled (for examples see Paisitkriangkrai *et al*. [24]) and some errors in the normalized DSM [23].

4. Approaches

This section introduces the components of our approach to high-resolution image segmentation in remote sensing. We first describe the first stage components, namely the patch-based pixel classification and the pixel-to-pixel segmentation, which allow us to achieve dense segmentation. We then elaborate on our idea of combining models to achieve improved overall segmentation accuracy, while preserving high classification performance for small classes, and introduce uncertainty maps.

4.1. Patch-based pixel classification

A CNN is trained on small image patches, which are extracted from the large training images. The patch size was chosen following the example of Paisitkriangkrai *et al*. [24], who achieved their best standalone CNN accuracy using 64×64 patches. However, we chose a 65×65 pixel shape, as we want to classify the image patch according to its center pixel. During the test-phase the trained CNN is used to classify the whole test image efficiently.

Architecture The chosen architecture for the patch-based CNN consists of four convolutional layers, followed by two fully connected layers. The first convolutional layer consists of 32 kernels of size $5 \times 5 \times 5$, which are applied with a stride of 1 on the $65 \times 65 \times 5$ input image. The second convolutional layer takes the output of the first convolutional layer as input and has 64 kernels of size $5 \times 5 \times 32$. There are 96 kernels of size $5 \times 5 \times 64$ in the third and 128 kernels of size $5 \times 5 \times 96$ in the fourth convolutional layer. Each of the convolutional layers is followed by a ReLU non-linearity, batch normalization [17] and a 3×3 max-pooling layer. The max-pooling operations are applied using a stride of 1, thereby avoiding down-sampling and allowing for a high spatial resolution. Weight initialization was performed following He *et al*. [16]. The two final fully connected layers consist of 128 neurons each and are followed by dropout layers with a 50% drop probability. The final layer consists of a 5-way softmax layer.

Data augmentation The training and validation data was generated by first extracting a patch for every car with the car being centered. Then additional training data for the car class is generated by rotating each of the patches several times at random angles. No translation augmentation was used, since we want to achieve high spatial resolution. The other classes are sampled randomly from the images, such that the center pixel belongs to the class of interest. The same amount of training data was sampled from each class to achieve class balance.

Fully convolutional classification To allow efficient classification of larger images, the fully connected layers are converted to convolutional layers following the example of Sermanet *et al*. [29]. This avoids the computational complexity of performing a sliding window approach, where overlapping regions would lead to redundant computations, and allows the classification of arbitrary image sizes.

4.2. Pixel-to-pixel segmentation

Inspired by the FCN architecture [22], we design an architecture that allows end-to-end learning of pixel-to-pixel semantic segmentation. The network is trained in mini-batches on patches of 256×256 pixels. The patch size was chosen due to GPU memory considerations.

Architecture The CNN architecture of the FCN network used in this paper is inspired by Simonyan and Zisserman [30] and is shown in Figure 2. The architecture consists of four sets of two 3×3 convolutions (blue layers), each set separated by a 2×2 max pooling layer with stride 2 (red layers). All convolution layers have a stride of 1, except the first one, which has a stride of 2. The change in the first convolution layer is a design choice, which was mainly made due to limits in GPU memory during test phase when considering large images. As in the patch-based architecture, all convolutional layers are followed by a ReLU non-linearity and a Batch normalization [17] layer. Weights were again initialized according to He *et al*. [16]. The

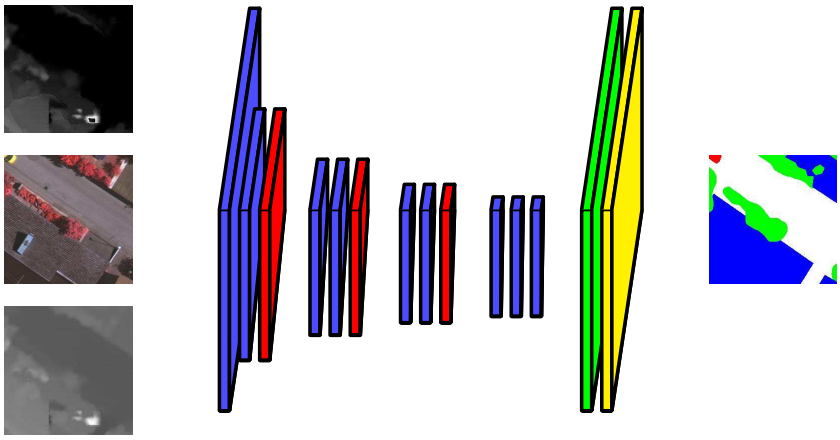

Figure 2: Pixel-to-Pixel architecture. Blue layers represent convolutional layers (including ReLU and batch-normalization layer), red layers represent pooling layers, the green layer represents the fractional-strided convolution layer and the yellow layer the softmax layer.

final 3×3 convolution is followed by a 1×1 convolution, which consists of one kernel for each class to produce class scores. The convolutional layers are followed by a fractional-strided convolution layer [22] (green layer, sometimes also referred to as deconvolution layer), which learns to up-sample the prediction back to the original image size and a softmax layer (yellow layer). The network is trained end-to-end using backpropagation.

Data augmentation The image patches are extracted from the input image with 50% overlap and are flipped (left to right and up down) and rotated at 90 degree intervals, yielding 8 augmentations per overlapping image patch.

Median frequency balancing Training of the FCN network was done using the cross-entropy loss function. However, as this loss is computed by summing over all the pixels, it does not account well for imbalanced classes. To take the imbalanced classes into account, two FCN models are trained: one using the standard cross-entropy loss, and one where the loss of the classes is weighted using median frequency balancing [5, 9]. Median frequency balancing weights the class loss by the ratio of the median class frequency in the training set and the actual class frequency. The modified cross-entropy function is

$$L = -\frac{1}{N}\sum_{n=1}^{N}\sum_{c=1}^{C} l_c^{(n)} \log\left(\hat{p}_c^{(n)}\right) w_c \,, \tag{1}$$

where N is the number of samples in a mini-batch,

$$w_c = \frac{\text{median}\left(\{f_c \mid c \in C\}\right)}{f_c} \tag{2}$$

is the class weight for class c, f_c the frequency of pixels in class c, $\hat{p}_c^{(n)}$ is the softmax probability of sample n being in class c, $l_c^{(n)}$ corresponds to the label of sample n for class c when the label is given in one-hot encoding and C is the set of all classes.

4.3. Uncertainty maps

Uncertainty maps are images that indicate the models uncertainty for a given pixel classification. Using Monte Carlo dropout [12] uncertainty maps are computed for all three CNNs by retrieving 10 Monte Carlo samples from the networks and then computing the standard deviation over the softmax outputs of the samples. In the rest of the paper, if not stated otherwise, we assume that uncertainty maps are displaying the mean standard deviation over all the classes. Besides the uncertainty maps, Monte Carlo sampling has also been shown to be valuable to increase classification accuracy [5, 12], as it has been shown to outperform the standard weight averaging technique.

4.4. Combined approach

Inspired by the idea of model ensembles, we propose a combination of the models to combine their strengths and achieve high overall accuracy, while still achieving high performance on small classes. We combine the softmax probabilities of the different methods by training one-vs-all linear SVMs on the combined softmax probabilities. (A linear SVM was chosen in order to speed up the training.) LIBLINEAR [10] was used to train the SVM.

5. Experiments and results

We evaluate the performance of the different methods on the ISPRS dataset with respect to overall accuracy and per-

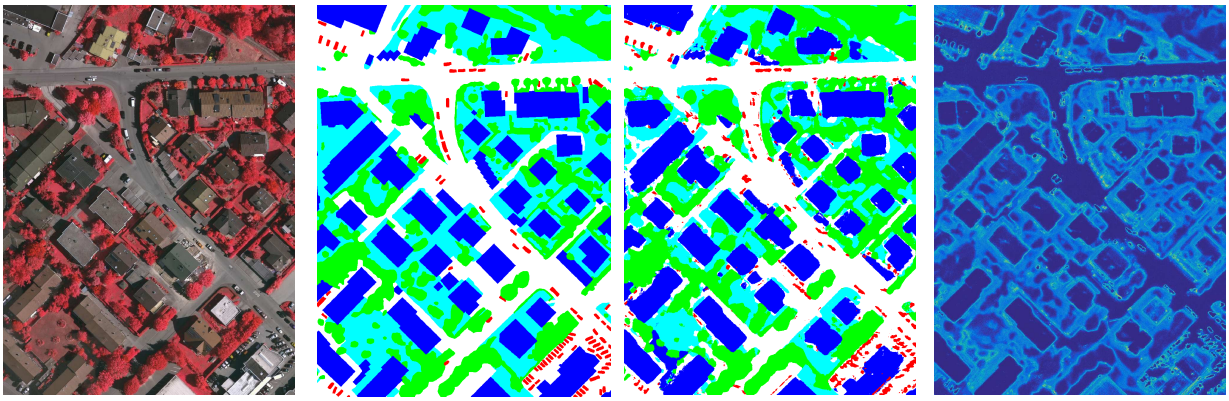

Figure 3: Results of the patch-based CNN. From left to right: One of the validation images, its ground truth, the results for the patch-based CNN and the uncertainty map.

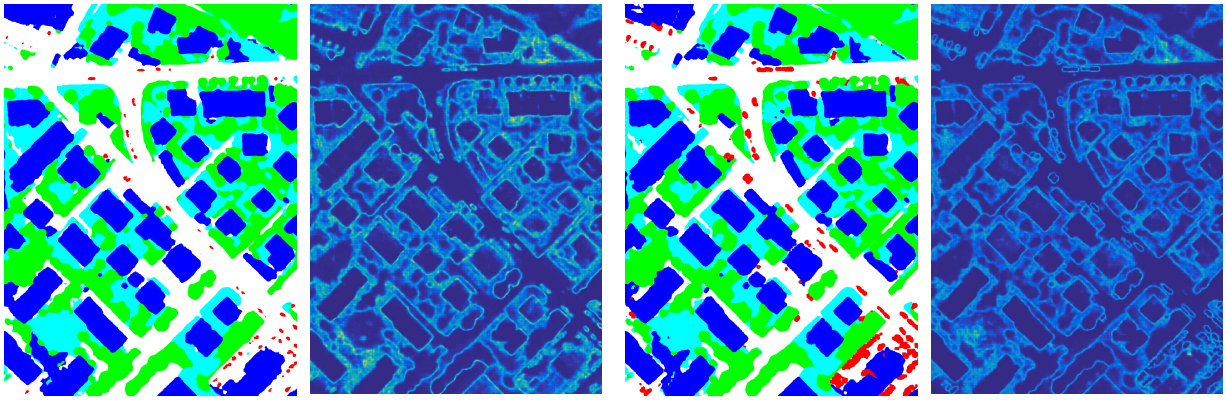

Figure 4: Results of the pixel-to-pixel CNN. From left to right: The results for the pixel-to-pixel CNN and the uncertainty map using the standard cross-entropy and using the cross-entropy with median frequency balancing, respectively.

formance on the small classes. Combinations of the methods are then considered to improve the stand-alone model performance. Finally, we evaluate the performance of the uncertainty measure.

Patch-based In this section we discuss the results for the patch-based CNN (PB). Figure 3 displays the classification results for one of the images in the validation dataset (area: 30). Note that due to the nature of the patch-based CNN architecture, classification of a 32 pixel boundary at all sides was omitted. It can be seen that the classification results are generally good for large objects, such as buildings and roads, however, there are many cars detected that are not actually in the image (false positives). Additionally, it can be observed that small areas of trees are misclassified as vegetation, or on rare occasions as buildings (bottom left corner). The uncertainty map displays the mean uncertainty over all the classes and it can be seen that the models uncertainty is especially high at boundaries and in regions where the model is performing misclassifications, for example in the bottom left corner. In semi-automatic approaches for remote sensing these areas could be presented to an operator for manual clarification. To visualize the uncertainty maps, the uncertainties were scaled into the range [0,1] and dark blue refers to low uncertainties, whereas lighter colors refer to higher uncertainties.

Table 1 shows the quantitative results for the patch-based CNN on the validation data set, and it can be seen that the worst class-accuracy is achieved for the car class.

Pixel-to-pixel The results of the pixel-to-pixel approach (FCN) for the same image as in Figure 3 is shown on the left side of Figure 4. Comparing its results to the patch-based approach illustrates the superior performance of the encoder-decoder architecture for segmentation. Edges of buildings are classified more evenly and small regions such as the trees at the bottom right side of the horizontal road are segmented out more accurately. This agrees with the quantitative results shown in Table 1, however, the quantitative results also illustrates that the native pixel-to-pixel approach

	Imp Surf		Building		Low veg		Tree		Car		Overall	
Method	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	Avg F1	Acc
PB	90.45	86.47	95.01	92.23	80.22	71.98	88.39	83.74	77.72	64.34	86.36	83.74
FCN	**92.77**	**89.96**	**95.81**	**94.10**	83.96	76.86	**90.81**	**87.39**	43.03	27.42	86.36	**86.65**
FCN-MFB	91.16	86.14	95.30	93.37	**84.36**	**77.80**	90.79	**87.39**	**92.57**	**86.61**	**90.84**	86.48

Table 1: Performance of the three different models. The F1 scores and accuracies are shown as percentages.

achieves worse results for the car class than the patch-based approach. From Figure 4 it can be seen that, compared to the patch-based approach, much fewer cars are detected and many car pixels are misclassified as roads (impervious surface).

Re-training the model with median frequency balancing (FCN-MFB) as described in Section 4.2, yielded much better accuracy for the car class, while still achieving a good overall classification accuracy close to the native FCN. The results can be seen on the right side of Figure 4 and in Table 1. Qualitatively the main difference to the standard FCN approach appears to be the increase in car classification frequency, however, this comes at the cost of some pixels being misclassified as cars.

The uncertainty information for both the FCN and the FCN-MFB are displayed in Figure 4. As for the patch-based approach it can be seen that the model uncertainty is quite high for boundary pixels, however, overall the FCN models appear to have very low uncertainty for most of the building pixels. Similar to the patch-based model it also gives high uncertainty for the vegetation region in the bottom left image corner.

Combined approach The results of combining the patch-based and the two pixel-to-pixel based approaches can be seen in Table 2. Comparing its results to Table 1, it can be seen that the total accuracy for the combined approaches is higher than for any of the single approaches. We illustrate results for all the combinations of the three approaches and observe that the best overall accuracy is achieved when combining all three models, whilst still achieving a good performance for the car class when comparing it to the best single model (FCN). The best accuracy for the car class is achieved when combining the patch-based approach and the median-frequency balancing FCN. However, this result comes at the cost of a decrease in overall accuracy. The segmentation result for our example image from the validation dataset when combining all three models can be seen in Figure 5. Overall, it can be seen that many of the misclassified cars from the patch-based method are removed. However, especially in the bottom right corner, it is possible to see that cars that are close together are merged into single blobs.

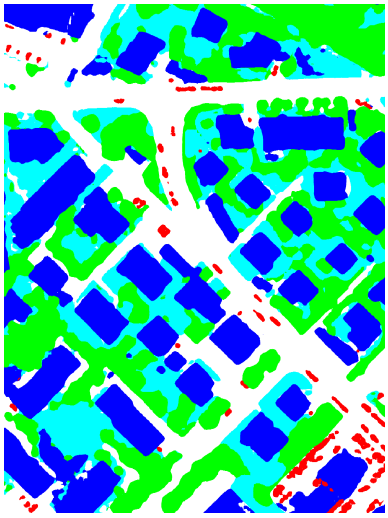

Figure 5: Result of the combined approach.

Uncertainty maps In this section we perform a novel analysis of uncertainty maps for CNNs in the context of remote sensing to illustrate that model uncertainty, the standard deviation of the Monte Carlo samples, is indeed related to classification accuracy. Figure 7 illustrates the relationship between uncertainty and accuracy. It displays the overall classification accuracy that is achieved on the validation dataset when dropping pixels that have a mean-class uncertainty above a certain threshold. The mean-class uncertainties have been normalized to the range $[0, 1]$ for all three models. Increasing the threshold and therefore including pixels with larger uncertainty leads to a decrease in overall accuracy for all three CNN approaches. This confirms our hypothesis that pixels with low uncertainty are more likely to be classified correctly.

Figure 6 illustrates, which pixels get included, when we threshold uncertainty such that the overall classification accuracy is at 97.5%, 95% and 90% for the FCN. It can be seen that the model is quite certain for many pixels and achieves a 97.5% accuracy, when including the pixels with least uncertainty. Here 64.85% of all the pixels in the image were classified, or when ignoring segmentation of the boundaries (as in the ISPRS contest) 67.65%. Increasing

	Imp Surf		Building		Low veg		Tree		Car		Overall	
Method	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	Avg F1	Acc
PB+FCN	**92.90**	**90.06**	**95.86**	**94.04**	83.81	76.54	90.98	87.83	54.94	37.89	83.70	86.84
PB+FCN-MFB	92.11	88.26	95.38	93.29	83.92	76.82	91.02	87.97	**83.62**	**71.97**	**89.21**	86.74
FCN+FCN-MFB	92.47	88.81	95.71	93.85	**83.98**	**76.87**	91.03	87.94	81.52	68.91	88.94	86.98
ALL	92.55	88.95	95.77	93.92	**83.98**	**76.87**	**91.05**	**87.99**	80.61	67.61	88.79	**87.03**

Table 2: Performance of the combined models. ALL refers to the combination of the PB, FCN and FCN MFB method.

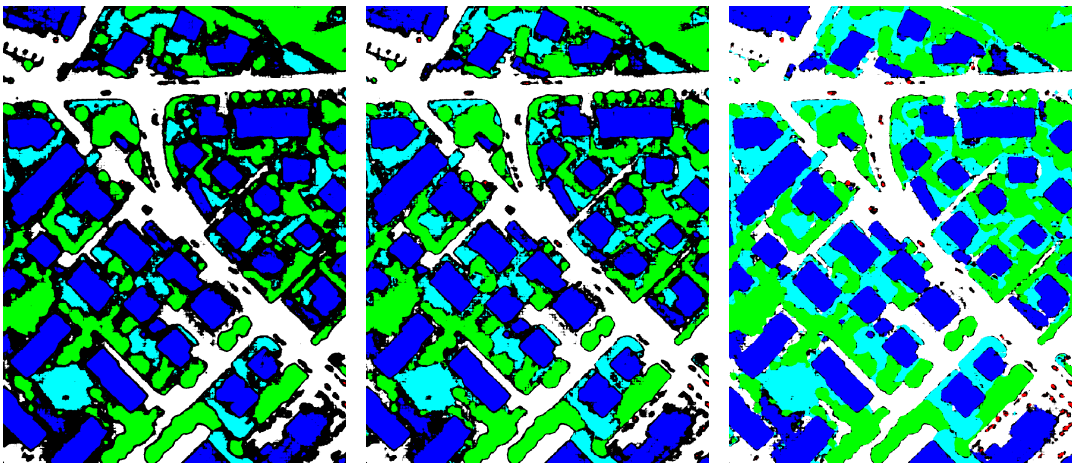

Figure 6: Results of the uncertainty experiment. The images show the segmentation for the FCN when setting the uncertainty threshold such that the overall accuracy is 97.5% (left), 95% (middle) and 90% (right). Black pixels are pixels that are not classified for a given threshold due to their uncertainty being larger than the defined threshold.

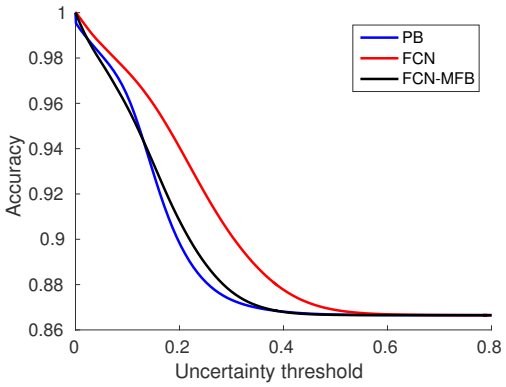

Figure 7: The relationship between accuracy and the uncertainty measure. We see that the accuracy decreases when more pixels with higher uncertainty are included.

the uncertainty threshold (middle Figure), a classification accuracy of 95% is achieved, while including 75.81% (or 78.27%) of the image pixels. When setting the threshold such that 90% accuracy is achieved, 94.06% (or 94.97%) of the pixels get classified. This illustrates that uncertainty maps can indeed be used to classify a large number of pixels with high accuracy.

Further, it can be seen that areas of class boundaries are the main cause of misclassification, which agrees with our previous observation that uncertainties were large in these areas. Additionally, it can be seen that the FCN model returns high uncertainties for cars and does barely segment any cars for the 97.5% and 95% threshold.

Experimental setup All experiments in this paper were performed using the deep learning framework Caffe [18] on a single Titan X, unless stated otherwise. To support segmentation of the large images in the patch-based approach, Caffe was modified to free redundant memory buffers during the forward pass during the testing stage. Additional modifications were made to support the median frequency balancing.

6. Discussion

The results showed that class imbalance may lead to reduced performance if not accounted for properly. This was evident when using the FCN approach, where the car class

was only classified with an accuracy of 27.4%. When optimizing the cost function for overall accuracy, classes with many pixels will automatically have a larger impact. Accordingly, more focus is put on improving the impervious surface class than the car class. This resulted in many cars being misclassified as roads. For the patch-based approach (car accuracy equal to 64.3%) this was not a problem, as classes were balanced in the training dataset. In this work we observed that median frequency balancing was essential to counter class imbalance when trying to perform segmentation of small objects using pixel-to-pixel CNNs. Incorporating median frequency balancing increased the accuracy of the FCN to 86.6% for the car class.

One region in our example image (area: 30) that contains many misclassified pixels for all the different approaches was the bottom left corner, where the models misclassified low vegetation as buildings. Figure 8 shows the RGB image and the normalized DSM for the small region of interest. It can be seen in the RGB image that the vegetation patch is elevated above the road and therefore separated from the road below by a strong edge. This can also be observed in the normalized DSM, where there is a drop both towards the road, but also towards the rest of the vegetation patch. This is a behavior, which we would usually expect for buildings and it is not surprising that the various approaches struggle with this particular area. However, the models generally display quite high uncertainty in that area (Figure 3 and 4), which confirms the importance of using uncertainty in scenarios where highly uncertain patches can be presented to an operator.

Smoothing the final pixel labeling results using a CRF has previously shown to increase overall classification accuracy [24] and could be combined with our approach. By using a CRF on the combined models we expect to achieve more robust classification results, in particular in terms of removing small mislabeled regions. It might, however, also impact the resolution of our segmentation by removing thin regions (e.g. pixels between parking cars).

Instead of learning CNNs from scratch, many CNNs have recently been based on the idea that CNN features are quite general and that CNNs can be pre-trained on large, often unrelated, datasets and can then be fine-tuned on the data at hand [4]. This allows for efficient training, even in situations where the available training dataset is small. One problem for remote sensing data is the fact that these large datasets and pre-trained networks generally only accept a three band input (RGB), which makes it inapplicable to situations where additional bands are available, such as in our scenario. Lagrange *et al*. [21] proposed to fine-tune separate CNNs for the different image bands and combine them using an SVM. A similar approach could be employed in our case and an investigation of this is left for future work.

As a continuation of this work, additional data sources (e.g. street view, satellite imagery) could be utilized to classify uncertain pixel areas in our segmentation, leading potentially to higher overall accuracy.

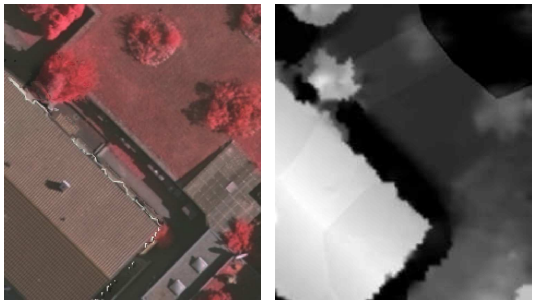

Figure 8: Zoomed in image of the bottom left corner of the image in Figure 3, where the vegetation area was misclassified as a building in all three models. The RGB image on the left and the normalized DSM on the right.

7. Conclusions

In this paper we have applied three recent approaches for pixel-wise classification based on advances in deep learning, and have analyzed their performance for small object segmentation and land cover mapping in urban remote sensing. We concluded that a combination of the models provided the best overall performance in terms of good accuracy for small objects, while still achieving a high overall accuracy. We also conclude that uncertainty maps, recently proposed by Gal and Ghahramani [12], are a good measure for the pixel-wise uncertainty of the segmented remote sensing images.

Acknowledgements We gratefully acknowledge the support of NVIDIA Corporation with the donation of the GPU used for this research and the German Society for Photogrammetry, Remote Sensing and Geoinformation (DGPF) for providing the Vaihingen data set [8]. This work was partially funded by the Norwegian Research Council FRIPRO grant no. 239844 on developing the *Next Generation Learning Machines*.

References

[1] ISPRS 2d semantic labeling contest. http://www2.isprs.org/commissions/comm3/wg4/semantic-labeling.html. 1, 2, 3

[2] S. Ardeshir, K. Malcolm Collins-Sibley, and M. Shah. Geo-semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2792–2799, 2015. 2

[3] S. Ardeshir, A. R. Zamir, A. Torroella, and M. Shah. Gis-assisted object detection and geospatial localization. In *Computer Vision–ECCV 2014*, pages 602–617. Springer, 2014. 2

[4] H. Azizpour, A. Razavian, J. Sullivan, A. Maki, and S. Carlsson. Factors of transferability for a generic convnet representation. 2014. 8

[5] V. Badrinarayanan, A. Kendall, and R. Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *arXiv preprint arXiv:1511.00561*, 2015. 4

[6] G. Camps-Valls, L. Bruzzone, et al. *Kernel methods for remote sensing data analysis*, volume 2. Wiley Online Library, 2009. 1

[7] F. Castaldo, A. Zamir, R. Angst, F. Palmieri, and S. Savarese. Semantic cross-view matching. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 9–17, 2015. 2

[8] M. Cramer. The dgpf-test on digital airborne camera evaluation–overview and test design. *Photogrammetrie-Fernerkundung-Geoinformation*, 2010(2):73–82, 2010. 8

[9] D. Eigen and R. Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2650–2658, 2015. 1, 4

[10] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. Liblinear: A library for large linear classification. *The Journal of Machine Learning Research*, 9:1871–1874, 2008. 4

[11] C. Farabet, C. Couprie, L. Najman, and Y. LeCun. Learning hierarchical features for scene labeling. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(8):1915–1929, 2013. 2

[12] Y. Gal and Z. Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. *arXiv preprint arXiv:1506.02142*, 2015. 1, 4, 8

[13] R. Girshick. Fast r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1440–1448, 2015. 2

[14] B. Hariharan, P. Arbelez, R. Girshick, and J. Malik. Hypercolumns for object segmentation and fine-grained localization. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 447–456, 2015. 1

[15] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015. 1

[16] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1026–1034, 2015. 3

[17] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015. 3

[18] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014. 7

[19] A. Kendall, V. Badrinarayanan, and R. Cipolla. Bayesian segnet: Model uncertainty in deep convolutional encoder-decoder architectures for scene understanding. *arXiv preprint arXiv:1511.02680*, 2015. 2

[20] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012. 2

[21] A. Lagrange, B. L. Saux, A. Beaupre, A. Boulch, A. Chan-Hon-Tong, S. Herbin, H. Randrianarivo, and M. Ferecatu. Benchmarking classification of earth-observation data: From learning explicit features to convolutional networks. In *Geoscience and Remote Sensing Symposium (IGARSS), 2015 IEEE International*, pages 4173–4176, 2015. 1, 2, 8

[22] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015. 1, 2, 3, 4

[23] I. Markus Gerke. Use of the stair vision library within the isprs 2d semantic labeling benchmark (vaihingen). 2, 3

[24] S. Paisitkriangkrai, J. Sherrah, P. Janney, and A. Hengel. Effective semantic pixel labelling with convolutional networks and conditional random fields. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 36–43, 2015. 1, 2, 3, 8

[25] O. A. B. Penatti, K. Nogueira, and J. A. dos Santos. Do deep features generalize from everyday objects to remote sensing and aerial scenes domains? In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 44–51, 2015. 1, 2

[26] P. H. Pinheiro and R. Collobert. Recurrent convolutional neural networks for scene parsing. *arXiv preprint arXiv:1306.2795*, 2013. 2

[27] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*, pages 91–99, 2015. 2

[28] A. B. Salberg. Detection of seals in remote sensing images using features extracted from deep convolutional neural networks. In *Geoscience and Remote Sensing Symposium (IGARSS), 2015 IEEE International*, pages 1893–1896, 2015. 1, 2

[29] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. *arXiv preprint arXiv:1312.6229*, 2013. 2, 3

[30] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 3

[31] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014. 1

[32] C. Szegedy, S. Ioffe, and V. Vanhoucke. Inception-v4, inception-resnet and the impact of residual connections on learning. *arXiv preprint arXiv:1602.07261*, 2016. 1

[33] S. Wang, S. Fidler, and R. Urtasun. Holistic 3d scene understanding from a single geo-tagged image. In *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on*, pages 3964–3972. IEEE, 2015. 2