Investigating patch-based and pixel-based deep architectures for dense semantic segmentation

M. Papadomanolaki[1 2], M. Vakalopoulou[2], K. Karantzalos[1]

[1] National Technical University of Athens
[2] CentraleSupélec, Université Paris-Saclay

mar.papadomanolaki@gmail.com

1 Abstract

Semantic segmentation is currently a mainstream method for addressing several remote sensing applications, achieving remarkable performance by employing deep learning techniques and more specifically, pixel-wise dense classification models. In this work, we present both quantitative and qualitative results of patch and pixel-based deep architectures for the semantic segmentation of different urban classes. The experiments are conducted using very high resolution images that are part of the ISPRS (WGIII4) benchmark dataset, depicting the city of Vaihingen.

2 Introduction

Semantic segmentation is a well studied problem for the remote sensing community. Traditionally, the approaches in the literature include supervised techniques, implementing different classifiers such as support vector machines or random forests and using a big variety of adhoc features, depending on the application, semantic categories and datasets. Additionally, sophisticated mathematical models such as Conditional Random Fields (CRF) or Markov Random Fields (MRF) were also used by semantic segmentation techniques to incorporate spatial relationships between objects [1, 2].

Currently, deep learning techniques and more specifically models which perform pixel-wise dense classification like fully convolutional networks (FCN), are holding the state-of-the-art results for semantic segmentation both in computer vision and remote sensing communities. Shelhamer et.al. [3] first proposed a FCN architecture for semantic segmentation problems, replacing the fully connected layers by convolutional layers with kernels that cover their entire input region. After this architecture a big variety of other architectures such as [4, 5, 6] have been proposed and reported very high accuracies on a wide range of applications.

In this work, we investigate and review patch-based deep architectures and compare them with pixel-based ones. More specifically, the explored patch-based models usually used in the literatures are: a relatively shallow ConvNet [8] and the AlexNet [7], in addition with two pixel-based architectures, namely SegNet [4] and Hourglass [6]. The Hourglass model is not usually employed for remote sensing classification as it was originally used to tackle the problem of human pose estimation. However, our aim is to inspect the output of such a model when dealing with dense semantic segmentation issues on satellite imagery.

3 Methodology

Next we describe in more detail the architectures and the layout of the experiments for the four employed models.

3.1 Patch-Based Architectures

We implement and compare two different conventional CNN architectures, one relatively shallow (ConvNet) and one relatively deeper (AlexNet). Here, by patch-based architectures, we mean that for each input patch, the model predicts only one label, which is assigned to the central pixel of the patch. Such a setting is quite inefficient as a patch is created for each pixel of the image, however patch-based methods are usually applicable in datasets with sparse annotations.

Concerning the employed models, ConvNet [8] is a relatively simple architecture which consists of 4 groups of layers: 2 convolutional and 2 fully-connected. Every convolutional layer is followed by a rectified linear unit (ReLU) function and a max pooling operation following the example of [8]. On the other hand AlexNet architecture follows the one of the original publication [7].

3.2 Pixel-Based Architectures

Concerning the pixel-based architectures, SegNet and Hourglass models had been implemented and compared. Both architectures produce dense predictions which means that for each input patch a map with per pixel class is generated.
SegNet architecture consists of 13 groups of layers similar to the ones of the VGG16 network, which have been initialized with the VGG16 pretrained weights. The entire architecture consists of repetitive blocks of convolutional, batch normalization, ReLU and indexed max-pooling layers, as described in the original publication [4].
On the other hand Hourglass is also based on the downsampling-upsampling idea but this time instead of convolution blocks the model includes several encoder-decoder parts that are successive and similar to each other. In this experimental setup we used four encoder-decoder parts. Similar to the SegNet architecture, the encoder-decoder block brings the input volume down to a very low resolution and then restores it back to the original dimensions using multiple residual modules. The term "residual" here is used owing to the fact that information is not just passed from one consecutive layer to another, but it is shared across multiple parts of the encoder-decoder block. Each residual layer consists of three layers which perform a batch normalization, a ReLU activation function and a convolution. Unlike other encoder-decoder approaches, this network does not make use of the common unpooling layers, but performs the upsampling alternatively using the nearest neighbor technique.

4 Implementation Details

4.1 Patch-Based Architectures

In order to train the ConvNet and AlexNet architectures, patches of size $29 \times 29 \times 3$ were extracted randomly taking 1% of each class from every image, resulting approximately in $1.100.100$ training and 38.000 validation patches. All the data was normalized by subtracting the mean and dividing by the standard deviation of the three available channels. Finally, the weights of both models were optimized by the standard Stochastic Gradient Descent with a learning rate of 0.04, a momentum of 0.9, a weight decay of 0.0005 and a batchsize equal to 100.
The architectures were trained on a GeForce GTX 980 GPU and implemented with the Torch open source library. Regarding ConvNet, 30 epochs of training were used lasting about 4 hours, while every 3 epochs the learning rate was decreased by subtracting the weight decay value. AlexNet needed approximately 20 hours and this time the model was trained for 60 epochs with the learning rate decreasing every 6 epochs.

4.2 Pixel-Based Architectures

Regarding SegNet, patches of size $256 \times 256 \times 3$ were extracted. More specifically, the patches were extracted using a step of 64 along both rows and columns forming overlapping small regions. Approximately 13800 training and 120 validation patches were created. All data were normalized before processed by the networks via mean and standard deviation. The same procedure was followed for Hourglass but this time with patches of size $128 \times 128 \times 3$ and a step of 54 because of memory constrains. Approximately 22000 training and 400 validation patches were created.
The pixel-based training tasks were assigned to the same GeForce GTX 980 GPU. Optimization was carried out by the Stochastic Gradient Descent for 60 epochs with the learning rate, momentum and weight decay values being equal to 0.01, 0.9 and 0.0005 respectively. The batchsize was equal to 10. For better performance, the learning rate was reduced by 10 after 25, 35 and 45 epochs. The whole implementation was completed in approximately 16 hours using the PyTorch platform. As far as the Hourglass is concerned, the training patches were feedforwarded to the model for 60 epochs. We employed the RMSprop optimization method, with a learning rate of $2.5e^{-}3$. The whole process lasted for about 24 hours on a single NVIDIA GeForce GTX TITAN with 12 GB of GPU memory.

5 Dataset

We chose the publicly available ISPRS Vaihingen 2D Labeling Challenge dataset for our experiments. It consists of 33 very high resolution images that depict the city of Vaihingen and have 3 available channels (InfraRed, Red, Green). 6 classes can be found in the image landscapes: Impervious Surfaces, Buildings, Low Vegetation, Trees, Cars and Clutter which represents everything else that is not included in the other five classes. 16 out of the 33 images are accompanied by groundtruth and the rest is intended for testing. From the 16 images that involve groundtruth information we used 14 for training and 2 for validation.

6 Experimental Results

In this section we present the evaluation of each one of the employed models. Both quantitative and qualitative results were produced via submission of our predicted ground-truth-less testing images to the ISPRS Test Project regarding Vaihingen 2D Labeling Challenge.

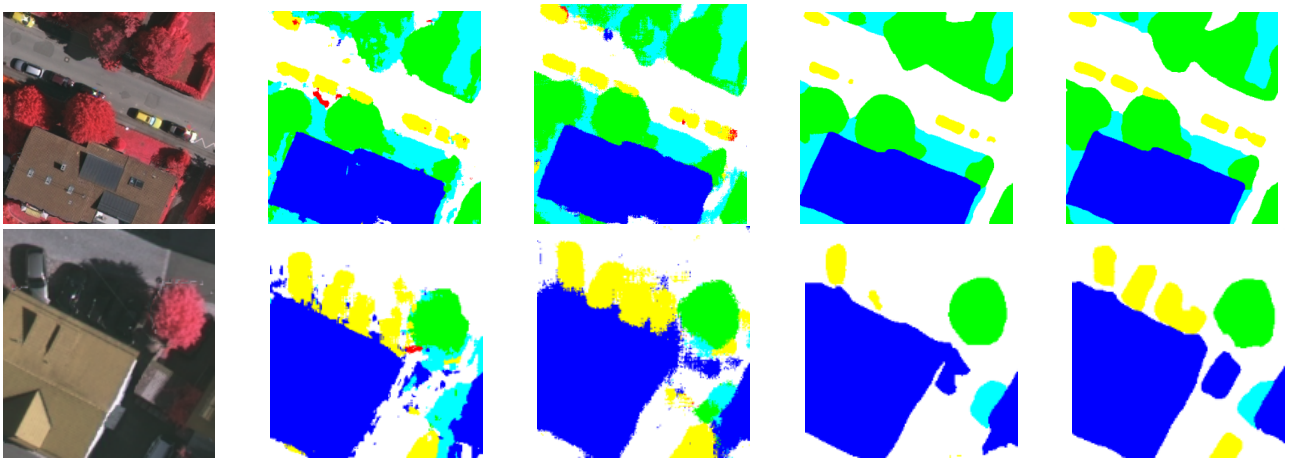

Figure 1: Zoomed in regions from areas #12 (top line) and #4 (bottom line): (from left to right) the original image, the predictions of ConvNet, AlexNet, SegNet and Hourglass are shown. *(White: Impervious Surfaces, Dark Blue: Buildings, Light Blue: Low Vegetation, Green: Trees, Yellow: Cars, Red: Clutter)*

6.1 Qualitative Evaluation

As observed in the quantitative evaluation, Hourglass locates cars with much more detail than SegNet. In Figure 1, a zoomed region of area 12 and area 4 of the ISPRS testing images is presented as an example. It is obvious that patch-based models produce more noisy images compared to the pixel-based ones. One can observe that the Hourglass architecture can detect cars even if they are in shadowed places with a very high accuracy. More specifically, for this specific testing area, the overall pixel-wise car accuracy that was extracted from the confusion matrix was 96%, outperforming the accuracy of SegNet which was 83%. In addition, the Hourglass model performs a more detailed detection of the building boundaries. Continuing with the bottom pictures of Figure 1, one can notice the difference between the two architectures on area 4 of ISPRS testing images. On the one hand, SegNet has merged the two neighboring building areas while on the other hand, Hourglass model has separated them as desired. Finally, the patch-based predictions produced by the ConvNet and AlexNet architectures are quite noisy, specially for the classes building and impervious surfaces (Figure 1 bottom).

6.2 Quantitative Evaluation

The quantitative analysis for ConvNet and AlexNet models is shown in Table 1. As expected, the accuracies of the ConvNet model (Table 1 left) are not very high since it is a relatively shallow architecture. Buildings are very often confused with impervious surfaces owing to their common spectral and spatial characteristics. Trees are also sometimes misclassified as low vegetation for the same reasons. Lastly, a relatively high accuracy has been achieved for cars if we consider the shallowness of the architecture. A similar situation prevails in the case of AlexNet (Table 1 right) only this time the accuracies are a little higher. The only exception is the category of impervious surfaces that has been associated more times with buildings. As far as total accuracies are concerned, AlexNet is again a little higher achieving 0.831 whereas ConvNet reaches 0.825.
As far as SegNet and Hourglass are concerned, the results are shown in Table 2. As one can observe, pixel-based classification approaches far outperform the patch-based framework both in per-category and in overall accuracy terms. Overall accuracy rates were equal to 0.886 and 0.873 for SegNet and Hourglass respectively. Comparing the two fully-convolutional architectures it is clear that they differ slightly. More precisely, SegNet detects buildings and low vegetation more successfully while the Hourglass network achieved better results regarding the categories of impervious surfaces, trees and cars. Lastly,

↓ predicted || reference →	imp_surf	building	low_veg	tree	car	clutter
imp_surf	**0.875**	0.042	0.060	0.007	0.012	0.003
building	0.120	**0.815**	0.046	0.004	0.011	0.003
low_veg	0.048	0.017	**0.753**	0.178	0.002	0.001
tree	0.009	0.001	0.122	**0.867**	0.001	0.000
car	0.096	0.069	0.031	0.003	**0.795**	0.006
clutter	0.269	0.357	0.065	0.005	0.093	**0.210**
Precision/Correctness	*0.836*	*0.919*	*0.736*	*0.827*	*0.506*	*0.485*
Recall/Completeness	*0.875*	*0.815*	*0.753*	*0.867*	*0.795*	*0.210*
F1	***0.855***	***0.864***	***0.744***	***0.846***	***0.618***	***0.293***

↓ predicted || reference →	imp_surf	building	low_veg	tree	car	clutter
imp_surf	**0.855**	0.059	0.048	0.010	0.023	0.003
building	0.095	**0.849**	0.034	0.004	0.015	0.002
low_veg	0.045	0.026	**0.753**	0.169	0.006	0.001
tree	0.007	0.002	0.119	**0.870**	0.002	0.000
car	0.043	0.077	0.011	0.003	**0.860**	0.005
clutter	0.198	0.364	0.041	0.007	0.100	**0.290**
Precision/Correctness	*0.859*	*0.897*	*0.762*	*0.832*	*0.390*	*0.580*
Recall/Completeness	*0.855*	*0.849*	*0.753*	*0.870*	*0.860*	*0.290*
F1	***0.857***	***0.872***	***0.758***	***0.850***	***0.537***	***0.387***

Table 1: Confusion matrixes for the patch-based models. In the left the confusion matrix of ConvNet is presented and on the right one from AlexNet.

↓ predicted || reference →	imp_surf	building	low_veg	tree	car	clutter
imp_surf	**0.933**	0.020	0.039	0.008	0.001	0.000
building	0.064	**0.908**	0.026	0.002	0.000	0.000
low_veg	0.038	0.012	**0.823**	0.127	0.000	0.000
tree	0.009	0.001	0.094	**0.896**	0.000	0.000
car	0.309	0.038	0.009	0.003	**0.633**	0.007
clutter	0.390	0.251	0.029	0.004	0.038	**0.287**
Precision/Correctness	*0.885*	*0.958*	*0.816*	*0.872*	*0.903*	*0.947*
Recall/Completeness	*0.933*	*0.908*	*0.823*	*0.896*	*0.633*	*0.287*
F1	***0.908***	***0.932***	***0.820***	***0.884***	***0.745***	***0.441***

↓ predicted || reference →	imp_surf	building	low_veg	tree	car	clutter
imp_surf	**0.947**	0.018	0.024	0.008	0.003	0.000
building	0.099	**0.878**	0.020	0.002	0.001	0.000
low_veg	0.068	0.016	**0.773**	0.143	0.000	0.000
tree	0.015	0.002	0.079	**0.904**	0.000	0.000
car	0.134	0.021	0.003	0.002	**0.840**	0.000
clutter	0.620	0.268	0.032	0.003	0.073	**0.004**
Precision/Correctness	*0.838*	*0.954*	*0.844*	*0.860*	*0.813*	*0.939*
Recall/Completeness	*0.947*	*0.878*	*0.773*	*0.904*	*0.840*	*0.004*
F1	***0.889***	***0.914***	***0.807***	***0.882***	***0.826***	***0.008***

Table 2: Confusion matrixes for the pixel-based models. In the left the confusion matrix of SegNet is presented and on the right one from Hourglass.

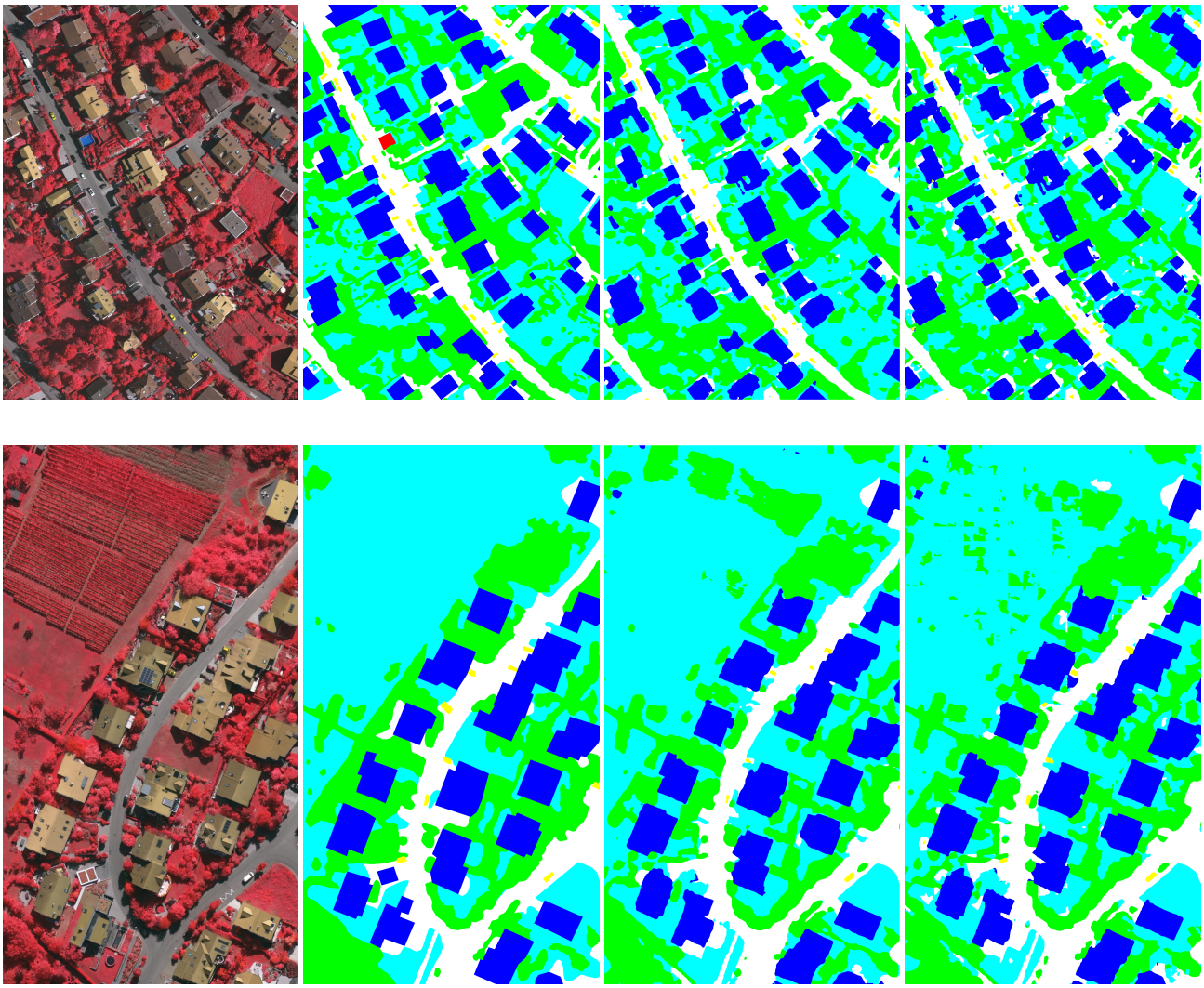

Figure 2: Images from the areas 15 (top line) and 17 (bottom line). From left to right: Satellite image depicting area 15 and area 17 of ISPRS dataset, the corresponding ground truth, the SegNet predictions, the Hourglass predictions.

the clutter category has a very low accuracy in SegNet results while it is almost not at all recognized by the Hourglass model resulting in a decrease of overall accuracy.

7 Conclusion

In this work, we tested both patch-based and pixel-based architectures for the semantic segmentation of very high resolution satellite images. As demonstrated by the evaluation of results, pixel-based models far outperform the patch-based ones. In addition, pixel-based architectures are much more flexible during inference time as it is not necessary to assign a prediction

to every pixel separately. SegNet achieves the highest accuracies, whereas the Hourglass architecture results have more correct shapes and more accurate borderlines between classes especially in the case of small objects such as cars. Using post processing techniques such as CRFs the proposed accuracies of the tested model can be further ameliorated. Finally, in the future we are planning to further evaluate the exploited architecture for the semantic instance segmentation problem.

References

[1] M. Vakalopoulou, N. Bus, K. Karantzalos, and N. Paragios, Integrating edge/boundary priors with classification scores for building detection in very high resolution data, *in 2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), July 2017,* pp. 3309–3312.

[2] M. Volpi and V. Ferrari, Semantic segmentation of urban scenes by learning local class interactions, *in 2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, June 2015, pp. 1–9.

[3] E. Shelhamer, J. Long, and T. Darrell, Fully convolutional networks for semantic segmentation, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 4, pp.640–651, April 2017.

[4] V. Badrinarayanan, A. Kendall, and R. Cipolla, Segnet: A deep convolutional encoder-decoder architecture for image segmentation, *IEEE PAMI, 2017.*

[5] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia, Pyramid scene parsing network, *in Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.*

[6] Alejandro Newell, Kaiyu Yang, and Jia Deng, Stacked Hourglass Networks for Human Pose Estimation, pp. 483–499, Springer International Publishing, Cham, 2016.

[7] Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton, ImageNet Classification with Deep Convolutional Neural Networks, *in Advances in neural information processing systems (NIPS), 2012*

[8] M. Papadomanolaki, M. Vakalopoulou, S. Zagoruyko, K. Karantzalos, Benchmarking Deep Learning frameworks for the classification of Very High Resolution Satellite Multispectral Data, *in ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Volume III–7, 2016