SateNet: Pixel-Level Segmentation of Satellite Images

Munachiso Nwadike
NYU Abu Dhabi Computer Science, 2019
msn307@nyu.edu

Abstract

Recent work in semantic segmentation has shown the effectiveness of context encoding in allowing neural networks to model spatial context for better performance. We set out to apply this development to the field of remote sensing. Our focus is to develop a new approach for semantic segmentation of Very High Resolution (VHR) images. This class of aerial images is often difficult to attain high quality semantic labels for, given the distinct challenges that come with images that have been remotely sensed via satellite or other aerial mechanisms. We make use of a lightweight architecture of convolutional neural networks, combined with context encoding to attain strong performance on the Vaihingen 2D semantic labeling competition. Our approach attains a competitive overall accuracy, as well as a strong classwise performance.

1. Introduction

Urban ecosystems have profound impact on the biogeochemical, climate and hydrological cycles of the environment in which they develop [1][2][3][4][5]. As urban sprawl increases, cities around the world have it in their best interest to monitor their landscapes to conduct environmental planning, economic forecasting [6][7][8], and, in the case of natural disaster, relief effort organisation [9][10][11]. One way to monitor ecosystems with modern technology is with remote sensing. Remote sensing is concerned with quantifying properties of objects located on the surface of the earth, using data acquired at a distance, usually through the use of a satellite or aircraft [12]. While remote sensing can focus towards collating sparse measurements over a geographical region in the form of point clouds [13], or working with time series measurements [14], we are largely concerned in this work, with remotely-sensed images. Specifically, we focus on the subset of remote sensing concerned with the understand of two-dimensional IRRGB grid, which has numerical values at each pixel location, without consideration for time dimension or depth channel.

Two-dimensional semantic labeling, or semantic segmentation, is one of the most active and oldest focus areas of computer vision research [15][16][16][17]. It focuses on assigning a class label, from a finite set of possible classes, to each pixel in an image or video [18]. This means effectively performing the related tasks of object detection and classification in one shot.

With improvements in storage and computing power, as well as ease of mining datasets from the internet, there has been a wealth of interest in semantic segmentation. A wealth of datasets have been released to provide a standard for research into this task [19][20][21][22][23]. We can formulate the semantic segmentation task as discriminative classification problem, in which we aim to learn the probability distribution $g_i = P(class = i|data)$, given the supervision of labels. While past techniques made use of hand-crafted features [24][19][25][26] to perform the segmentation task, modern methods have made use of deep learning. Deep Convolutional Neural Networks (DCNN), in particular, have dominated in achieving the state of the art in this area [27][28][29][30][31][32]. In the context of remote sensing, we may refer to the semantic segmentation task as landcover classification [33][34].

Achieving accurate semantic segmentation results from remotely-sensed images would seem to a be particularly difficult task. The quality of datasets, in terms of the resolution of aerial imagery, has only improved with advances in technology, as argued by [35]. Ground truth labels for pixels in remotely-sensed images thus becoming more consistent with their real world interpretation. This in turn demands more fine grained classification of pixels from semantic segmentation algorithms. However, the nature of such images is that while their spatial resolution is high, their spectral resolution is low [33]. They also inherently contain many many man-made objects [36][37], which are made of different substances. Some of these objects objects are very small in size, and can interfere with overall quality of classification. These challenges are added to the usual ones faced in semantic segmentation such as dealing with occlusions and shadows, and learning feature invariance.

Yet, Deep Convolutional Neural Networks have been particularly successful in achieving state of the

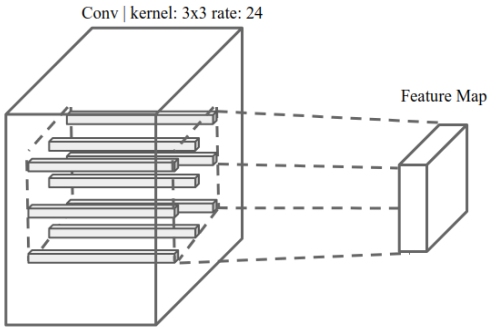

Figure 1. Atrous convolution using kernel size 3x3 with a rate of 24. These are used within the context encoding module.

art semantic segmentation of remotely-sensed images [38][39][40][41][42]. As has been argued by [33], the strength of neural networks comes from their ability to learn end-to-end mappings, eliminating the need for handcrafted features. The hierarchical nature of convolutions allows for the output value of a single pixel location to be influenced by a large region in the input image. However, downsampling is a necessary part of these convolutions, and creates a degree of information loss, which reduces the performance of the neural network. There have been many attempts to mitigate this with innovations such as atrous convolution, visualised in figure 1 [43][44][45].

The problem with atrous convolution is that it prevents pixels from taking information in the global scene context into account. Recent work from [32] has argued precisely this, and shows that the use of context encoding module used with a convolutional neural network (CNN) as the backbone, can allow for state of the art accuracy on general purpose image segmentation, while leveraging information from the global scene context. The purpose of our work is to investigate the effect of combining the context encoding approach with existing methods in remote sensing, and to see how this impacts the quality of remotely-sensed image segmentation. To this end, we got inspiration from recent work on [46], which showed the viability of a simple encoder-decoder architecture for segmentation of urban remotely-sensed data in achieving state of the art accuracy. One of the primary datasets for testing the architecture in that work was Vaihingen dataset from the ISPRS 2D Image segmentation task [37]. For completeness, we note that this dataset was also the focus of this paper.

2. Previous works

In 2010, Mnih etal. had the first successful application of patch-based CNN to detect roads and buildings from remotely-sensed images [47]. They developed a custom Massachusetts Roads Dataset on which they had trained their CNN. However, despite their breakthrough using CNNs, the usage of handcrafted features in semantic segmentation of aerial images continued for some time. Paisitkriangkrai et al. [48], in 2015, used a combination of CNN features, and hand-crafted features to achieve accuracy of 88% on the ISPRS semantic labeling challenge. They used a CNN together with a random forest (RF) classifier, and applied a conditional random field to refine the combined CNN and RF probablities. In the same year, however, Castelluccio et al. [49] were able to make use of a pretrained GoogLeNet in achieving state of the art in the well-known UC-Merced dataset. Their focus task was to identify, for any given aerial image, what the land in the image was for in terms of its zoning assignment. The difficulty of this task comes from the similar ways in which land may be used in different parts of a city, for different designations. For example, a dense residential area is distinct from a medium residential area, though these categories may be hard to distinguish depending on the scale of the images representing them.

In 2015, Marmanis et al. [50] showed that by using a simple CNN originally designed to handle the ImageNet challenge, they were able to increase overall accuracy on the UCMerced Land Use dataset from the previously reported best score of 83.1% to 92.4%.

SegNet [51] was developed in 2015, as a relatively memory and computation-efficient encoder-decoder architecture for performing general image segmentation. In 2016, Audebert et al. [52] showed that taking advantage of the simple architecture of SegNet, they were able to achieve state of the art performance on the Vaihingen Image Segmentation Challenge. They proposed many different variants of SegNet for processing of Very High Resolution remotely-sensed images. Their best performing variant of SegNet, however, made use of a fusion of two versions of SegNet. One of these made use of depth maps (DSM) provided in the Vaihingen dataset, while the other made use of plain IRRG images.

Marcu et al.[53], in 2016, made use of a modified VGG-net to process local image patches, and a modified AlexNet to process global image patches, before combining the outputs of these two distinct networks with a fully connected layer to give the segmentation prediction result. While their work was tested on their own custom datasets, they were also able to improve upon the results of Mnih et al. on the Massachusetts dataset . Also in 2016, Marmanis et al. [54] made use of an ensemble of CNNs, specifically VGG-16 pretrained on ImageNet, FCN-Pascal pretrained on Pascal VOC, and Places pretrained on the MIT Places dataset. They finetuned these netwokrs on the Vaihingen 2D labeling dataset, and were able to achieve accuracy of 88.5%, which at the time was consistent with the state of the art.

In 2017, Liu et al.[55] made use of a novel Self-Cascaded Convolutional Neural Network (ScasNet) to achieve state of the art performance on the Vaihingen dataset. Their pro-

posed method was able to make use of context in multiple scales, which is has an advantage over the use of multiple images, of being more efficient. He et al. [56], in 2015, had introduced the concept of residual networks, which make use of skip connections to speed up convergence and increase the accuracy of the neural network. However, in 2017, Zhang et al. [57] combined this architecture with that of U-Net [58], to demonstrate state of the art performance in road network detection on the dataset developed by Mihn et al.

Marmanis et al. [59] proposed, in 2018, to improve segmentation quality of SegNet by introducing a boundary detector to prevent the blurring of boundaries. Theirs is a combination work that combines segnet with a modified boundary detector from *Holistically-Nested Edge Detection* [60]. Their work achieved a new state of the art accuracy on the Vaihingen benchmark.

Recent work from Yang et al.[61] in 2019, showed that a version of ResNet, altered at the level of individual residual blocks[62], in combination with the spatial pyramidal pooling module of [63] was able to perform aerial image segmentation. Their approach did not achieve state of the art in the Vaihingen dataset, but showed competitive performance in the related Potsdam semantic labeling dataset.

Some exciting work has also been done with semi-supervised Generative Adversarial Networks (GAN), which Kerdegari et al. [64] used to achieve a competitive overall accuracy on the Vaihingen and Potsdam dataset. This work, from 2019, though recent, is distinct form our work, as it makes use of research in generative modeling.

Work from the University of Toronto [65] showed that by incorporating a CNN to predict energy maps, they are able to more efficiently minimise an energy function which provides the final image segmentation. The model showed competitive performance on the Vaihingen dataset, as well as on a custom TorontoCity dataset of aerial city images.

While the focus of this paper is on remote sensing images of a location generated at a fixed point in time, interesting work is also being done in the related research area of change detection. Change detection, in remote sensing, is focused on specifying the changes that take place in an aerial image over time at fixed points [66]. J. Liu et al., showed that through a dept redesign of its loss function, U-Net could be made to achieve state of the art performance on the SZTAKI AirChange Benchark dataset [67]. Since change detection involves and understanding of the bi-temporal relationship between a 2D image of the locations in question, R. Liu et al. [68] made use of a BiLSTM [69] to model these relationships. They first extracted the image features using a CNN, and made use of the attention mechanism [70] to improve the performance of the LSTM.

Finally, a paper released in 2019, which has a similar motivation to our paper, comes from Mou et al. [71]. Much like our paper, they use an established network, VGG-16, as a backbone to extract a set of feature maps from input images. They then make use of special modules, in their case called the spatial relation and channel relation modules, to respectively extract global spatial and contextual dependencies from these feature maps. Their works obtained competitive accuracy on the Vaihingen dataset.

3. Method Description

3.1. SegNet

SegNet is computationally inexpensive. It has just 5 convolutional blocks in its encoder and decoder respectively. The encoder of SegNet consists of the first 13 convolutional layers from VGG16 network [72]. We may group these convolutional layers into 5 convolutional blocks, each of which has two or three convolutions, and batch normalisation layers, with ReLU. At the end of each convolutional block, there is a maxpooling layer. The convolutional layer has kernels of size 3x3, and a stride of 1, while the maxpooling layers in the encoder are of size 2x2. The fully convolutional layers of VGG16 are note used, and instead, the output of feature maps from the

The decoder of SegNet is essentially a mirror of the encoder. There are two or three convolutions, along with batch normalisation and relu, right after unpooling layers. The unpooling are the inverse of the maxpooling layers, and are of size 2x2. This is demonstrated in figure 4. Maxpooling allows for invariance in small shifts in features that may occur within the output feature map.

The encoder-decoder architecture of SegNet is particularly favourable for semantic segmentation of aerial images, since we require the target to be at the same scale as the input, given how sensitive the images are to distance.

Given that the encoder makes use of downsampling via maxpool, there is an inherent loss of information from each convolutional block in the encoder. Specifically, while the use of downsampling may allow the network to be more invariant to translation of the input image, there is a loss of spatial resolution that from the input image, and from feature maps produced by convolutional layers in the network. MaxPool is not an invertible operation, since non-maximal values are set to zero within each pooling window. However, we can compute a partial inverse to maxpooling in the unpooling layer. To achieve this, we simply save the indices of the maximum value from each window in the maxpooling layer, in the form of a pooling mask. We then use this mask to situate relevant values in the corresponding unpooling layer of the decoder. This allows us to preserve some of the spatial relationship and context between features. As we will discuss in our section on skip connections, the use of downsampling and reupsampling, even with aid of the pooling mask, still means that some information is lost from the

feature maps, since pixel values are being completely reset to zero, and the output of the unpooling is a sparse feature map. This is partially compensated by the fact that we increase the number of feature maps output in from the max-pooling and unpooling layers, so that more parameters can be learned.

The outputs of the network are normalised using a softmax function before being passed through the cross entropy loss function. The mathematical representation of this step is shown in figure 1. Note that we do not pass the feature maps directly into the softmax to directly predict probabilities, but first pass them through the context encoding module, which we will discuss in the next section.

$$A = \frac{1}{M}\sum_{j=1}^{M}\sum_{k=1}^{N} y_k^j log(\frac{e^{x}{}_k^j}{\sum_{i=1}^{N} e^{x}{}_i^j}) \quad (1)$$

In this equation, j is the specified pixel, y^j is its label, and (x_1^j, x_2^j, x_M^j) is the input vetor we are normalising in the last layer. In this context, we have N classes and k possible segmentation classes.

We observe that while SegNet has an Encoder-Decoder architecture, it is also fully convolutional. This means that it does not have the fully connected layers which are responsible for much of the computational resource consumption of a deep convolutional neural network. We have experimented with another backbone feature extractor in the form of DeepLabv3+, and were not able to see any improvement compared to SegNet, and hence found SegNet to be a sufficient basis for further experiments.

3.2. Context Encoding Module

We make use of a modified context encoding module, originally introduced by [32]. Our version of the context encoding module makes use of skip connection.

The goal of the context encoding module is to better help in utilizing information from the global context. To achieve this, the context encoding module must receive a set of feature maps, which represent features extracted from an input image, as its input. As a result, it was a natural choice for us to add it to the decoder part of the SegNet architecture.

The context encoding module operates by learning two sets of learnable parameters. The first is a set of codewords, which we use to calculate a code residual i.e the norm distance between a set of features and the codewords, and a scaling factor, with which we scale the residuals. We then aggregate the code residuals to obtain a final image feature of size $c * 1 * 1$ from the context encoding module.

The image feature which servers as our encoding then goes to serve two purposes. On the one hand, it is used in computing the semantic encoding loss. Much like a per-pixel loss, semantic loss forces the context encoding module to make use of semantic context. However, it regularises the module by allowing both large and small object categories to have equal semantic relevance. On the other hand the encoding feature is broadcast over the original context encoding input feature maps, causing it to be resized to the size of the original feature map, having coded the relevant context information. This is illustrated by our diagram of the overall architecture show in figure 2

A picture of our overall architecture can be seen in figure 2

Within the context encoding module, we make use of many convolutions and batch normalisation layers. In the third and fourth stage of convolution, we specifically make use of atrous convolutions, demonstrated in figure 1.

Upon running experiments, we noticed that the context encoding module was adding too much memory overhead to run on our available GPU, so we made use of a modified version of the context encoding module. We first downsampled the input to the context encoding module, which causes a loss of information, so we make use of a skip connection which we add to the upsampled output of the context encoding module. We discuss skip connections in more detail in later sections.

3.3. Skip Connections

Initially we had a problem fitting the model, with the additional context encoder on GPU. We decided to modify the aggregation of learnable codewords to ignore the weighting of code residuals. However, we realised that this amounts to collect information from previous layers of the network into the context encoding module and essentially teaching the model not to make use of this learned information. In practise, this was affecting the accuracy of the model, limiting the testing accuracy on the Vaihingen dataset to approximately 84%. However, we eventually realised that so save computational power, we can simply downsample the feature maps passed into the convolution layer as this is allows for a less memory intensive computations. This means that the codewords and scale factors are still learned by the context encoding module. However, the use of downsampling in the context encoding module mandated that we use upsampling after learning the codewords and feature weights in the context encoding module, since we need to maintain a consistency between the dimensions of the output feature map with the target images. Without an increase in the number of output channels generated from the maxpooling, it was clear that this downsampling and upsampling would cause some loss of information in the network. We realised that drawing on the works of He et al. [56] we could make use of skip connections to allow for information to be passed from the input of the context encoding module to its output.

While this was purely an architectural decision, the use of skip connections in the context encoding module proved

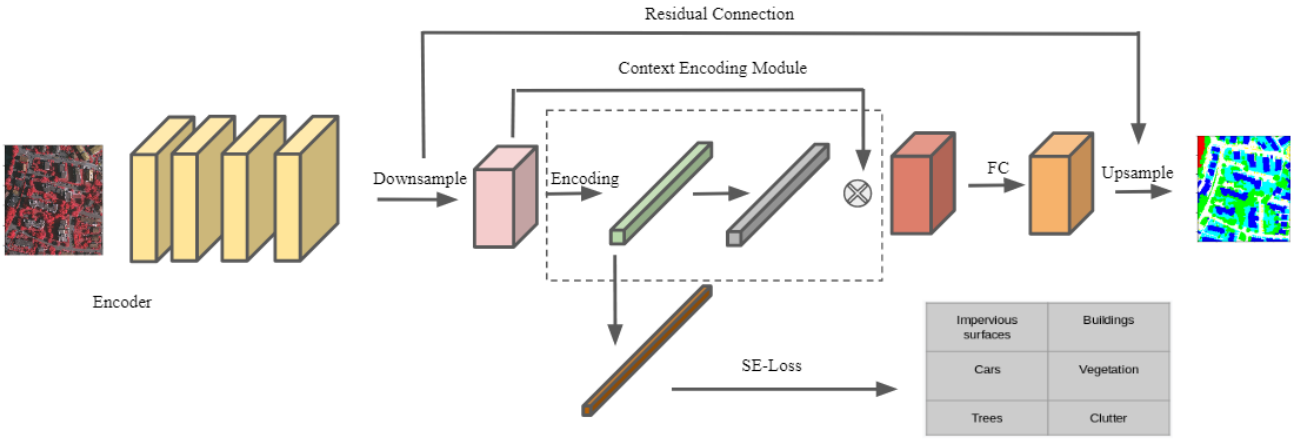

Figure 2. Diagram of our architecture. The output of the SegNet architecture is passed to the modified context encoding module, from which the semantic segmentation output is given.

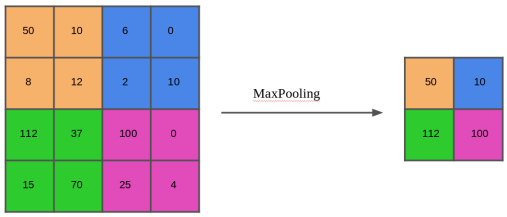

Figure 3. Visualisation of two-dimensional maxpooling used to downsample the input. There is a degree of information loss created by downsampling, which we try to compensate for by passing information via skip connection, demonstrated in figure 5.

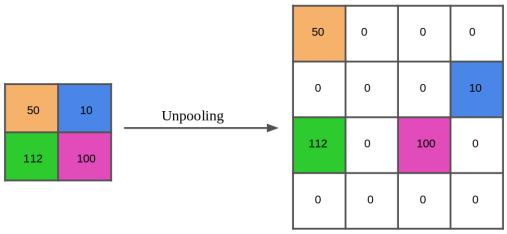

Figure 4. Unpooling is necessary after dowsampling the image for processing through the context encoding module. The non-invertible nature of downsampling operations means that unpooling does not allow us to regain all information lost in downsampling.

to be a wise decision. Assuming that the goal of the context encoding layer is to learn a target function $F(x)$, by adding the input of the context encoding module to its output via skip connection, we force it to model a new function $h(x) = F(x) - x$, while the signal of $F(x)$ is already passing through it. This means that the addition of our modified context encoding module allows for faster convergence than the ordinary context encoding module, while being less memory intensive.

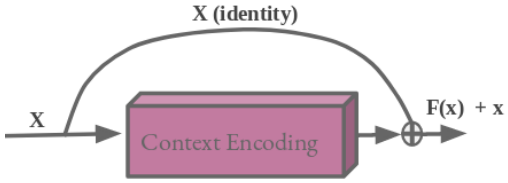

Figure 5. We made use of skip connection[56] to maintain information lost from downsampling at the beginning of the context encoding module.

4. Experiments

4.1. Vaihingen Dataset

The Vaihingen Dataset consists of 33 image patches cropped from a true orthophoto(TOP) mosaic. Figure 6 shows how the image patches were generated.

The labeling for the images is provided at the pixel level in the form of an image, wherein each pixel is in the color corresponding to its label. Each image has its own labeling image, called its *ground truth*. There are seven categories of roads, buildings, low vegetation, trees, cars, clutter, and Undefined categories, respectively represented by the colors white, blue, cyan, green, yellow, red and black. In evaluation, we only pay mind to the F1 scores for first five categories. There is also a version of the image labels, referred to as *eroded grond truth*, in which the boundaries of objects are eroded by a circular disc of 3-pixel radius. Those eroded areas are then ignored during evaluation. The motivation is to reduce the impact of uncertain border definitions on the evaluation. Using this ground truth we are able to obtain higher performance, and hence focused on this eroded ground truth in practice. Specifically, while you cannot train on the eroded labels, you can deploy them in

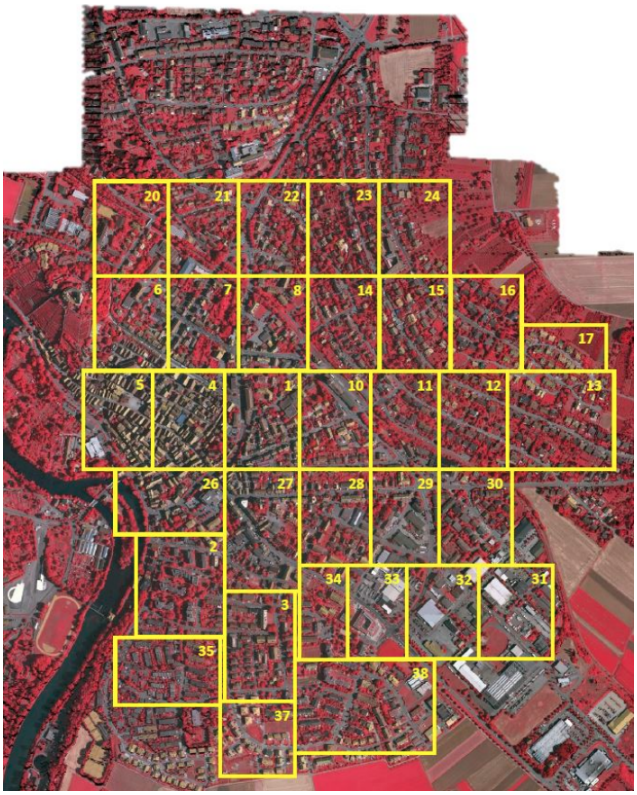

Figure 6. The Vaihingen dataset consists of 33 image patches cropped from an aerial image of the city urban city of Vaihingen Germany

testing.

Each of the images in the Vaihingen dataset has dimensions in the width and height of 2100 pixels, with a resolution of approximately 9cm per pixel. The TOP images consist of the near-infrared, red, and green (IRRG) bands instead of ordinary RGB channels. Each image in the dataset also has a corresponding Digital Surface Model (DSM) which is a similarly sized image in which each pixel provides information on the elevation of a given point in the terrain, given by Lidar point cloud. For the purposes of our project, we did not make use of information from DSM. A sample of image patch number 17 from the Vaihingen dataset is shown in figure 7.

4.2. Implementation details

Our algorithm was built using the open source PyTorch tool [73]. We trained for 40 epochs on our dataset. While the work of SegNet made use of ordinary stochastic gradient descent, we were not able to use it to achieve for convergence of our algorithm. Hence, we made use of the Adam optimizer, originally proposed by [74].

In training, we take random patches of the training images, to allow the neural network to learn better variety. It also allow us to speed up the training process. In testing,

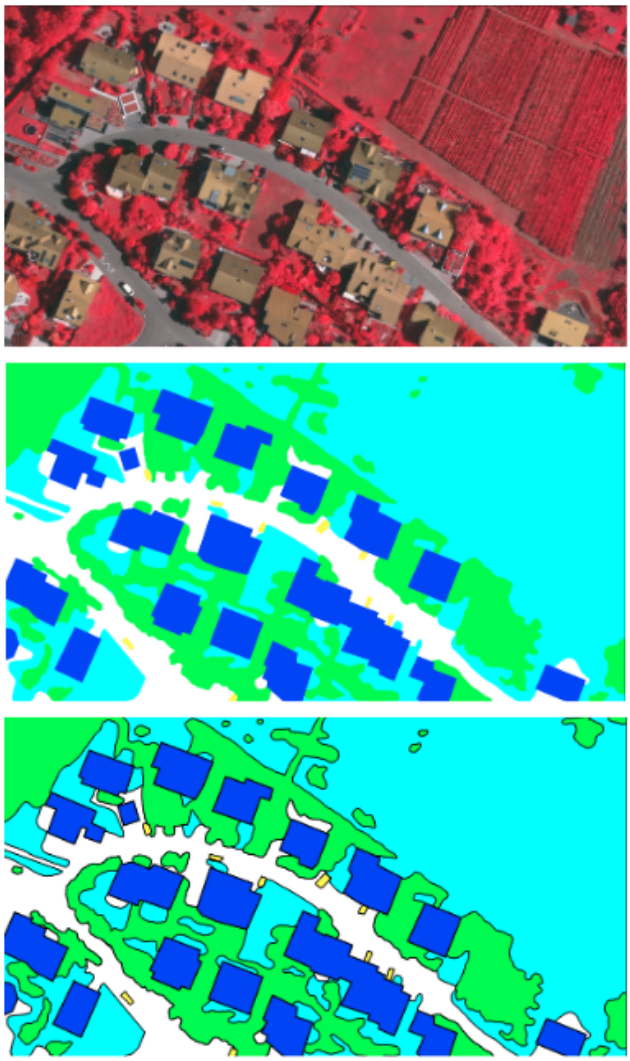

Figure 7. The uppermost image shows the TOP view of image patch 17 from the Vaihingen dataset, compared with its ground truth label, which comes second, and finally, with the corresponding eroded ground truth label.

however, the input images are concatenated into one long image feature, before being forward passed through the network to generate the segmentation predictions. We made use of a batch size of 4, and train on an NVIDIA GeForce GTX GPU using CUDA toolkit.

4.3. Evaluation

We evaluated the model using F1score, classwise F1 score, and Overall Accuracy. To get a better understanding of the specific areas where the model is making misclassification, we produce the diagram in table 1, which shows the confusion matrix between the various classes. As expected, the highest number lie on the diagonal, indicating that most of the classifications are done correctly.

	Predicted					
Actual	Roads	Buildings	Low Veg	Trees	Cars	Clutter
Roads	4390256	159957	174598	65265	13108	1181
Buildings	152307	5198191	61557	10198	689	5
Low Veg	159819	62787	2292570	387275	259	0
Trees	40678	9497	321029	4021480	76	0
Cars	24748	5915	95	749	116137	1033
Clutter	0	0	6216	0	0	0

Table 1. Confusion matrix, Showing the Frequency with which the various categories are confused As expected, , most categories are classified correctly. The network does confuse certain similar categories like Low Vegetation and Trees, and fewer numbers of dissimilar categories.

Method	Roads	Buildings	Low Veg	Trees	Cars	OA
FCN[35]	90.5	93.7	83.4	89.2	72.6	89.1
SegNet(IRRG)[46]	91.5	94.3	82.7	89.3	85.7	89.4
SegNet-RC[46]	91.0	94.5	84.4	89.9	77.8	89.8
V-FuseNet[46]	91.3	94.3	84.8	89.9	**86.3**	90.0
NLPR3*	**93.0**	95.6	**85.6**	90.3	84.5	**91.2**
SateNet(Ours)	91.7	**95.7**	79.6	**90.6**	83.3	90.6

Table 2. Table showing the results on the Vaihingen dataset. Other than Overall Accuracy (OA), all values taken represent the F1 score. **NLP3 paper not released.*

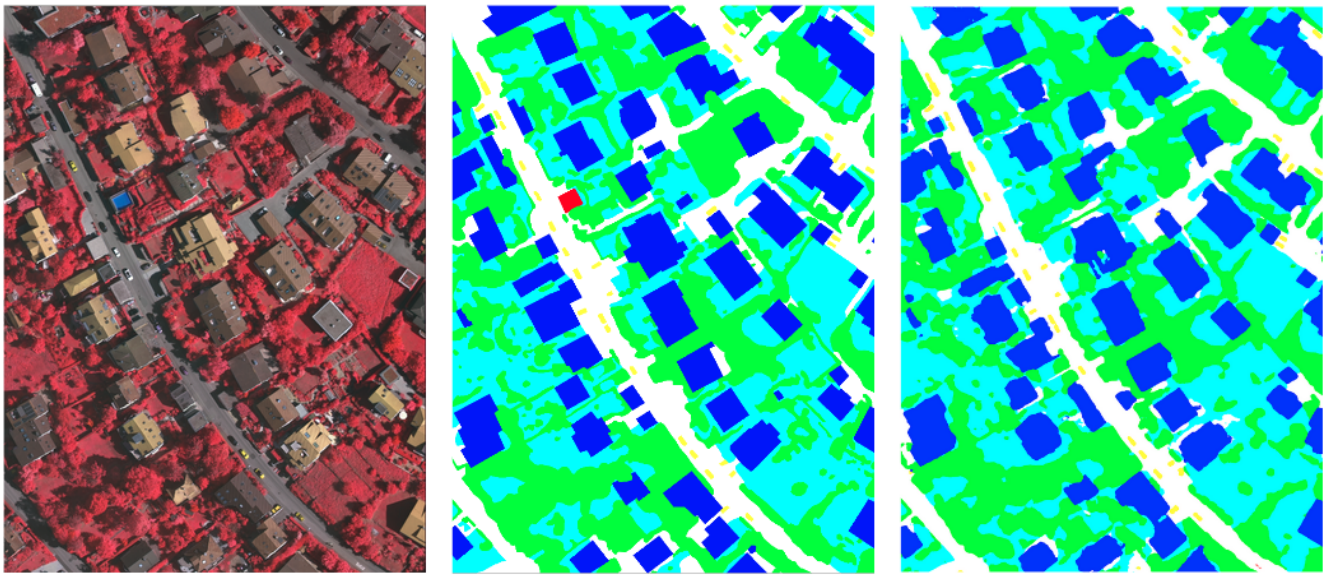

Figure 8. Image showing the ground truth, compared to what our model predicts, versus what can be achieved with regular SegNet architecture. That particular image is patch 15 from the dataset.

Our model compares well with other models, as can be seen from table 2. As shown, at the time of writing this paper, the state of the art is NLPR3, which had an overall accuracy of 91.2. While their results have been relesed on the ISPRS website, we cannot cite them as they are yet to release their complimentary paper detailing their method.

In order to improve the model performance, we also attempt to use dropout in the layers of the feature extractor/encoder-decoder. Dropout was introduced by Hinton et al. in [75] as way to prevent a neural network from overfitting by selectively excluding some of the neurons in the network on each pass. However, we found that the use of dropout resulted in a 20% decrease in model performance. This appears to be because the size of our network is already small relative to the size of the dataset, so partially lowering its effective size in various training epochs hinders its ability to generalize effectively.

We also attempted to decrease the size of our network by removing 20% of the convolutional blocks in the encoder-decoder. However, we saw no significant improvements to our evaluation results. This seemed to confirm our thoughts about the use of dropout, as it is similar in effect to reducing the already network size.

References

[1] Nancy B. Grimm, Stanley H. Faeth, Nancy E. Golubiewski, Charles L. Redman, Jianguo Wu, Xuemei Bai, and John M. Briggs. Global change and the ecology of cities. *Science*, 319(5864):756–760, 2008.

[2] Marina Alberti. The effects of urban patterns on ecosystem function. *International Regional Science Review*, 28(2):168–192, 2005.

[3] Michael P Johnson. Environmental impacts of urban sprawl: A survey of the literature and proposed research agenda. *Environment and Planning A: Economy and Space*, 33(4):717–735, 2001.

[4] G. Darrel Jenerette and Larissa Larsen. A global perspective on changing sustainable urban water supplies. *Global and Planetary Change*, 50(3):202 – 211, 2006.

[5] Marjorie van Roon. Water localisation and reclamation: Steps towards low impact urban design and development. *Journal of Environmental Management*, 83(4):437 – 447, 2007.

[6] S. T. A. Pickett, M. L. Cadenasso, J. M. Grove, C. H. Nilon, R. V. Pouyat, W. C. Zipperer, and R. Costanza. Urban ecological systems: Linking terrestrial ecological, physical, and socioeconomic components of metropolitan areas. *Annual Review of Ecology and Systematics*, 32(1):127–157, 2001.

[7] Marina Alberti. Urban patterns and environmental performance: What do we know? *Journal of Planning Education and Research*, 19(2):151–163, 1999.

[8] Karen C Seto and J Marshall Shepherd. Global urban land-use trends and climate impacts. *Current Opinion in Environmental Sustainability*, 1(1):89 – 95, 2009.

[9] Christine Wamsler. Mainstreaming risk reduction in urban planning and housing: A challenge for international aid organisations. *Disasters*, 30(2):151–177, 2006.

[10] S. Voigt, T. Kemper, T. Riedlinger, R. Kiefl, K. Scholte, and H. Mehl. Satellite image analysis for disaster and crisis-management support. *IEEE Transactions on Geoscience and Remote Sensing*, 45(6):1520–1528, June 2007.

[11] I. Z. Gitas, A. Polychronaki, T. Katagis, and G. Mallinis. Contribution of remote sensing to disaster management activities: A case study of the large fires in the peloponnese, greece. *International Journal of Remote Sensing*, 29(6):1847–1853, 2008.

[12] R.A. Schowengerdt. *Remote Sensing: Models and Methods for Image Processing*. Elsevier Science, 2006.

[13] Xiang Li, Mingyang Wang, Congcong Wen, Lingjing Wang, Nan Zhou, and Yi Fang. Density-aware convolutional networks with context encoding for airborne lidar point cloud classification, 2019.

[14] Oumer S. Ahmed, Steven E. Franklin, Michael A. Wulder, and Joanne C. White. Characterizing stand-level forest canopy cover and height using landsat time series, samples of airborne lidar, and the random forest algorithm. *ISPRS Journal of Photogrammetry and Remote Sensing*, 101:89 – 101, 2015.

[15] Claude R Brice and Claude L Fennema. Scene analysis using regions. *Artificial intelligence*, 1(3-4):205–226, 1970.

[16] Edward M Riseman and Michael A Arbib. Computational techniques in the visual segmentation of static scenes. *Computer Graphics and Image Processing*, 6(3):221–276, 1977.

[17] Azriel Rosenfeld and Larry S Davis. Image segmentation and image models. *Proceedings of the IEEE*, 67(5):764–772, 1979.

[18] R. Szeliski. *Computer Vision: Algorithms and Applications*. Texts in Computer Science. Springer London, 2010.

[19] Jamie Shotton, John Winn, Carsten Rother, and Antonio Criminisi. Textonboost for image understanding: Multi-class object recognition and segmentation by jointly modeling texture, layout, and context. January 2009.

[20] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba. Scene parsing through ade20k dataset. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5122–5130, July 2017.

[21] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. *CoRR*, abs/1604.01685, 2016.

[22] Roozbeh Mottaghi, Xianjie Chen, Xiaobai Liu, Nam-Gyu Cho, Seong-Whan Lee, Sanja Fidler, Raquel Urtasun, and Alan Yuille. The role of context for object detection and semantic segmentation in the wild. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[23] Holger Caesar, Jasper R. R. Uijlings, and Vittorio Ferrari. Coco-stuff: Thing and stuff classes in context. *CoRR*, abs/1612.03716, 2016.

[24] J. Yao, S. Fidler, and R. Urtasun. Describing the scene as a whole: Joint object detection, scene classification and semantic segmentation. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 702–709, June 2012.

[25] Stephen Gould, Richard Fulton, and Daphne Koller. Decomposing a scene into geometric and semantically consistent regions. pages 1 – 8, 11 2009.

[26] L. Ladický, C. Russell, P. Kohli, and P. H. S. Torr. Associative hierarchical crfs for object class image segmentation. In *2009 IEEE 12th International Conference on Computer Vision*, pages 739–746, Sep. 2009.

[27] Zhenli Zhang, Xiangyu Zhang, Chao Peng, Xiangyang Xue, and Jian Sun. Exfuse: Enhancing feature fusion for semantic segmentation. In *The European Conference on Computer Vision (ECCV)*, September 2018.

[28] Chenxi Liu, Liang-Chieh Chen, Florian Schroff, Hartwig Adam, Wei Hua, Alan L. Yuille, and Li Fei-Fei. Auto-deeplab: Hierarchical neural architecture search for semantic image segmentation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[29] Zilong Huang, Xinggang Wang, Lichao Huang, Chang Huang, Yunchao Wei, and Wenyu Liu. Ccnet: Criss-cross attention for semantic segmentation. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019.

[30] P. Wang, P. Chen, Y. Yuan, D. Liu, Z. Huang, X. Hou, and G. Cottrell. Understanding convolution for semantic segmentation. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1451–1460, March 2018.

[31] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. *CoRR*, abs/1802.02611, 2018.

[32] Hang Zhang, Kristin J. Dana, Jianping Shi, Zhongyue Zhang, Xiaogang Wang, Ambrish Tyagi, and Amit Agrawal. Context encoding for semantic segmentation. *CoRR*, abs/1803.08904, 2018.

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

[34] K. Chen, K. Fu, M. Yan, X. Gao, X. Sun, and X. Wei. Semantic segmentation of aerial images with shuffling convolutional neural networks. *IEEE Geoscience and Remote Sensing Letters*, 15(2):173–177, Feb 2018.

[35] Jamie Sherrah. Fully convolutional networks for dense semantic labelling of high-resolution aerial imagery. *CoRR*, abs/1606.02585, 2016.

[36] 2d semantic labeling contest - potsdam.

[37] 2d semantic labeling - vaihingen data.

[38] Sakrapee Paisitkriangkrai, Jamie Sherrah, Pranam Janney, and Anton van den Hengel. Effective semantic pixel labelling with convolutional networks and conditional random fields. In *CVPR Workshops*, pages 36–43, 2015.

[39] M. Kampffmeyer, A. Salberg, and R. Jenssen. Semantic segmentation of small objects and modeling of uncertainty in urban remote sensing images using deep convolutional neural networks. In *2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 680–688, June 2016.

[40] W. Sun and R. Wang. Fully convolutional networks for semantic segmentation of very high resolution remotely sensed images combined with dsm. *IEEE Geoscience and Remote Sensing Letters*, 15(3):474–478, March 2018.

[41] Xu Ji, João F. Henriques, and Andrea Vedaldi. Invariant information distillation for unsupervised image segmentation and clustering. *CoRR*, abs/1807.06653, 2018.

[42] Ronald Kemker, Carl Salvaggio, and Christopher Kanan. Algorithms for semantic segmentation of multispectral remote sensing imagery using deep learning. *ISPRS Journal of Photogrammetry and Remote Sensing*, 145:60 – 77, 2018. Deep Learning RS Data.

[43] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *CoRR*, abs/1606.00915, 2016.

[44] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *The European Conference on Computer Vision (ECCV)*, September 2018.

[45] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *CoRR*, abs/1706.05587, 2017.

[46] Nicolas Audebert, Bertrand Le Saux, and Sébastien Lefèvre. Beyond rgb: Very high resolution urban remote sensing with multimodal deep networks. *ISPRS Journal of Photogrammetry and Remote Sensing*, 140:20 – 32, 2018. Geospatial Computer Vision.

[47] Volodymyr Mnih and Geoffrey E. Hinton. Learning to detect roads in high-resolution aerial images.

[48] Sakrapee Paisitkriangkrai, Jamie Sherrah, Pranam Janney, and Anton Hengel. Effective semantic pixel labelling with convolutional networks and conditional random fields. pages 36–43, 06 2015.

[49] Marco Castelluccio, Giovanni Poggi, Carlo Sansone, and Luisa Verdoliva. Land use classification in remote sensing images by convolutional neural networks. *CoRR*, abs/1508.00092, 2015.

[50] Dimitris Marmanis, Mihai Datcu, Thomas Esch, and Uwe Stilla. Deep learning earth observation classification using imagenet pretrained networks. *IEEE Geoscience and Remote Sensing Letters*, 13:1–5, 12 2015.

[51] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[52] Nicolas Audebert, Bertrand Le Saux, and Sébastien Lefèvre. Semantic segmentation of earth observation data using multimodal and multi-scale deep networks. *CoRR*, abs/1609.06846, 2016.

[53] Alina Marcu and Marius Leordeanu. Dual local-global contextual pathways for recognition in aerial imagery. *CoRR*, abs/1605.05462, 2016.

[54] Dimitrios Marmanis, Jan D Wegner, Silvano Galliani, Konrad Schindler, Mihai Datcu, and Uwe Stilla. Semantic segmentation of aerial images with an ensemble of cnns. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 3:473, 2016.

[55] Y. Liu, B. Fan, L. Wang, J. Bai, S. Xiang, and C. Pan. Context-aware cascade network for semantic labeling in vhr image. In *2017 IEEE International Conference on Image Processing (ICIP)*, pages 575–579, Sep. 2017.

[56] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.

[57] Zhengxin Zhang, Qingjie Liu, and Yunhong Wang. Road extraction by deep residual u-net. *CoRR*, abs/1711.10684, 2017.

[58] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. *CoRR*, abs/1505.04597, 2015.

[59] Dimitrios Marmanis, Konrad Schindler, Jan Dirk Wegner, Silvano Galliani, Mihai Datcu, and Uwe Stilla. Classification with an edge: Improving semantic image segmentation with boundary detection. *CoRR*, abs/1612.01337, 2016.

[60] Saining Xie and Zhuowen Tu. Holistically-nested edge detection. *CoRR*, abs/1504.06375, 2015.

[61] Haiping Yang, Bo Yu, Jiancheng Luo, and Fang Chen. Semantic segmentation of high spatial resolution images with deep neural networks. *GIScience & Remote Sensing*, 56(5):749–768, 2019.

[62] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. *CoRR*, abs/1603.05027, 2016.

[63] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia. Pyramid scene parsing network. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6230–6239, July 2017.

[64] Hamideh Kerdegari, Manzoor Razaak, Vasileios Argyriou, and Paolo Remagnino. Urban scene segmentation using semi-supervised GAN. In Lorenzo Bruzzone and Francesca Bovolo, editors, *Image and Signal Processing for Remote Sensing XXV*, volume 11155, pages 464 – 471. International Society for Optics and Photonics, SPIE, 2019.

[65] Dominic Cheng, Renjie Liao, Sanja Fidler, and Raquel Urtasun. Darnet: Deep active ray network for building segmentation. *CoRR*, abs/1905.05889, 2019.

[66] ASHBINDU SINGH. Review article digital change detection techniques using remotely-sensed data. *International Journal of Remote Sensing*, 10(6):989–1003, 1989.

[67] J. Liu, K. Chen, G. Xu, X. Sun, M. Yan, W. Diao, and H. Han. Convolutional neural network-based transfer learning for optical aerial images change detection. *IEEE Geoscience and Remote Sensing Letters*, pages 1–5, 2019.

[68] R. Liu, Z. Cheng, L. Zhang, and J. Li. Remote sensing image change detection based on information transmission and attention mechanism. *IEEE Access*, 7:156349–156359, 2019.

[69] Mike Schuster and Kuldip Paliwal. Bidirectional recurrent neural networks. *Signal Processing, IEEE Transactions on*, 45:2673 – 2681, 12 1997.

[70] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[71] Lichao Mou, Yuansheng Hua, and Xiao Xiang Zhu. A relation-augmented fully convolutional network for semantic segmentation in aerial scenes. *CoRR*, abs/1904.05730, 2019.

[72] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[73] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

[74] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[75] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958, 2014.