Deep Autoencoders with Aggregated Residual Transformations for Urban Reconstruction from Remote Sensing Data

Timothy Forbes, Charalambos Poullis
Immersive and Creative Technologies Lab
Department of Computer Science and Software Engineering
Concordia University
Montreal, Canada
timforby@gmail.com, charalambos@poullis.org

***Abstract*—In this work we investigate urban reconstruction and propose a complete and automatic framework for reconstructing urban areas from remote sensing data.**

Firstly, we address the complex problem of semantic labeling and propose a novel network architecture named SegNeXT which combines the strengths of deep-autoencoders with feed-forward links in generating smooth predictions and reducing the number of learning parameters, with the effectiveness which cardinality-enabled residual-based building blocks have shown in improving prediction accuracy and outperforming deeper/wider network architectures with a smaller number of learning parameters. The network is trained with benchmark datasets and the reported results show that it can provide at least similar and in some cases better classification than state-of-the-art.

Secondly, we address the problem of urban reconstruction and propose a complete pipeline for automatically converting semantic labels into virtual representations of the urban areas. An agglomerative clustering is performed on the points according to their classification and results in a set of contiguous and disjoint clusters. Finally, each cluster is processed according to the class it belongs: tree clusters are substituted with procedural models, cars are replaced with simplified CAD models, buildings' boundaries are extruded to form 3D models, and road, low vegetation, and clutter clusters are triangulated and simplified.

The result is a complete virtual representation of the urban area. The proposed framework has been extensively tested on large-scale benchmark datasets and the semantic labeling and reconstruction results are reported.[1]

***Keywords*-component; formatting; style; styling;**

I. Introduction

Object recognition is today one of the most researched topics in the field of computer vision. Accurately and automatically labeling the objects in an image presents a challenge that is progressively seeing better results, especially with the use of deep learning techniques. At the same time classification of geospatial objects has gained considerable traction primarily due to the wide availability of benchmark datasets for 2D and/or 3D semantic labeling. Most of these datasets consist of remote sensing data of multiple large-scale urban areas with annotated ground truth.

In this work we address the problems of (a) semantic labeling of geospatial objects from remote sensing data, and (b) automatic urban reconstruction based on semantic labels.

Semantic labeling. We investigate and propose the use of convolutional autoencoders with feed-forward feature map links where each convolutional layer consists of a block with aggregated residual transformations known as ResNeXT [26] and we show how this network achieves smooth predictions while retaining high frequency information and can produce similar and in some cases better classification than state-of-the-art. For training the network we use the ISPRS (International Society for Photogrammetry and Remote Sensing) benchmark dataset [21] and in particular a dataset of an urban area from a historical city in Germany (Vaihingen). The data is in the form of depth map generated using structure-from-motion/multi-view stereo-based techniques, and high resolution orthophotos. The geospatial objects present in the data are buildings, roads, trees, low vegetation, cars, and clutter. Once trained, the network is used to label 17 testing images for which no ground truth was provided which are evaluated against a set of known metrics.

Urban reconstruction. We propose a pipeline for the reconstruction of urban areas based on semantic labeling. The pipeline is fully automated and includes clustering the points based on their label and specialized processing for each of the labels of geospatial objects. In particular, trees, cars, buildings, roads, low vegetation, and clutter are converted into virtual representations of the urban area.

Our technical contributions are:

- a complete framework for the geospatial object classification and reconstruction of large-scale urban areas.
- the design, development of a novel network architecture for supervised learning. We show how the network can perform at least similar and in some cases better than state-of-the-art networks in terms of classification accuracy.
- a method for automatically converting semantic labels

[1]Renderings and videos of the results can be downloaded from here

of geospatial objects into 3D models and in particular, how models for generic objects such as trees, cars, and buildings can be generated as part of the immersive virtual worlds.

Paper Organization. The paper is organized as follows: Section II presents an overview of the state-of-the-art in the areas of semantic labeling and urban reconstruction. In Section III we present a technical overview of our proposed technique and in Section IV we provide a brief description of the dataset used. The proposed architecture is described in detail in Section V including the training, validation and comparisons with state-of-the-art. Section VI presents the proposed pipeline for the automatic reconstruction of urban areas based on the semantic labeling of the area and experimental results are shown in Section VII. The conclusion and future work are discussed in Section VIII.

II. Related Work

There is a vast body of work in semantic labeling and 3D reconstruction. Below we provide a brief overview of state-of-the-art related work in terms of (a) semantic labeling, and (b) urban reconstruction.

A. Semantic Labeling

The problem of semantic labeling has been explored through a variety of methods. Prior to deep learning, semantic labeling relied on hand engineered features. One of these methods proposed the generation of features that were classified into unary potentials then fed into conditional random fields (CRF), localizing the label and segmenting object instances [23].

Deep learning, over the past few years, has proved to be very effective at object recognition due to its ability to learn important features. Applying deep learning to semantic segmentation becomes a challenge as localized information is often lost in favor of high-level information. Chen et al [6, 7] apply a CRF or a discriminatively trained domain transform to the model's output to preserve edge information and to smooth semantic segmentation. Noh et al [16] perform deconvolution to reach the original input resolution allowing the network to learn localization through deconvolution kernels.

Other works have the deep network perform the segmentation by preserving low-level information for the network's segmentation process. Long et al [14] combine low-level information from their pooling layers to their final layers. SegNet [2], the network that inspired our work, consists of encoders and decoders that share pooling indices in order to preserve lower level information. Several other works apply this concept with slight derivations. The authors in [18] keep a single residual stream with information at the original resolution. One network consists of holding previous pixel-specific layer activations within vectorized columns [9]. Another uses visual and geometric cues during unpooling [8], while [12] uses separate network paths to capture all available information from earlier layers. A few high-performing networks base their work off the idea that a convolution has less contextual reach than assumed and they employ larger kernels [17], global image information [13], different pooled feature maps [4, 27].

Variants of the aforementioned network architectures have been employed in the context of semantic labeling of geospatial features each with unique advantages and trade-offs [10]. In this work, we propose a novel network architecture that combines the strengths of different networks which when used for semantic labeling on remote sensing data can achieve similar and in some cases better classification accuracy than state-of-the-art.

B. Urban Reconstruction

Urban reconstruction has been an active research area since the early 80s hence it is no surprise that a vast body of work exists. A comprehensive survey of state-of-the-art can be found in Musialski et al [15] where techniques proposed over the past few years are categorized according to the objective, type, and scale of data. In this section, we provide a brief overview of state-of-the-art in large-scale urban reconstruction from remote sensing data, most relevant to our work.

There are techniques which use symmetries and regularities in the geometry. Zhou et al [28] proposed an automated system which given the *exact bounding volume of a building* can simplify the geometry based on dual contouring while retaining important features. Using this technique the authors were able to simplify the original geometry considerably. On a similar line of research, Lafarge et al. [24] proposed a method which produces excellent reconstructed models from pointcloud data which can also produce models at different levels-of-detail. Although techniques produce impressive results, they do require considerable user interaction during the pre-processing stage typically in the form of carefully identifying the objects' points.

Other techniques aim for full-automation and are therefore entirely data-driven. On such example is the work of Poullis et al [20] where pointcloud data is converted automatically to polygonal 3D models. This technique is applicable directly on the raw pointcloud data without requiring any user interaction. Later, in [19] the authors extended the work to include a fast boundary refinement algorithm based on graph-cuts which was used to refine the boundaries. Overall, these techniques scale well with vast amounts of data however this comes at the cost of increasing the difficulty of enforcing symmetry constraints such as the Manhattan world assumption. In other words, larger areas can be processed but the generated models are noisier than the previous approach of using regularities. A solution to this side-effect was proposed by Arikan et al [1] where a system for generating polyhedral models from semi-dense

unstructured point-clouds was developed. Planar surfaces were first extracted automatically based on prior semantic information, and later refined manually by an operator.

Finally, a rather different approach was proposed by Xiao et al [25] where the authors used inverse constructive solid geometry to address the reconstruction problem. Rather than using boolean operations on simple primitives to generate a complex structure, they start off with a point cloud representing the indoor area of a structure and decompose that into layers which are then grouped into higher-order elements. This works very well for highly regularized scenes such as indoor spaces however it does not produce useful results for large-scale outdoor areas.

To conclude, the weakest link in all reported work in urban reconstruction is the geospatial object classification: a misclassified object causes wrong reconstruction results since all subsequent steps are dependent on the classification. Furthermore, extracting buildings from LiDAR data often produces jagged boundaries which affects the accuracy and quality of the reconstruction. Similarly, correctly classifying trees and in particular trees which are taller and overhanging on buildings allows for better reconstruction. Thus, it is of outmost importance that the classification is as accurate as possible at the pixel-level. The proposed neural network achieves this yielding average accuracy in the high ninetieth percentile for buildings and trees, as well as similar with or better than the accuracy with state-of-the-art for the other classes.

III. System Overview

The data input is two geo-registered remote sensing images; an orthophoto in the form of IR-RG (InfraRed-Red,Green) and a corresponding depth map. The images are classified using the proposed SegNeXT which produces smooth predictions while retaining high frequency details. The SegNeXT predictions are then used as a proxy in grouping the 3D points into disjoint and contiguous clusters. Finally, to reconstruct the scene each cluster is processed depending on its classification: for buildings the boundaries are extracted and 3D polygonal models are extruded, trees are replaced by procedural models and their placement is determined by a Voronoi tessellation of the cluster, cars are replaced by simplified CAD models, and roads, low vegetation, and clutter are replaced by a simplified mesh of the terrain. Figure 1 shows a diagram of the pipeline of the proposed framework.

IV. Dataset

The training and testing of the network is performed on data provided by the International Society for Photogrammetry and Remote Sensing (ISPRS) as part of a benchmark competition on urban object detection and 3D building reconstruction [21]. Multiple datasets are available for each competition i.e. 2D semantic labeling, 3D semantic labeling, 3D reconstruction, etc. Our objective is 2D semantic labeling thus we have used the Vaihingen dataset which consists of a total of 33 pairs of IRRG (infrared, red, green) images and their associated depth maps. Each image is an orthophoto of the historical city Vaihingen, Germany, has an average resolution 2000×2500, and a sampling density of $9cms$. For each IRRG-depth image pair, a ground-truth image is also provided showing the manually assigned per-pixel classification into six classes: (a) buildings (blue), (b) roads (white), (c) trees (green), (d) red (clutter), (e) low vegetation/natural ground (cyan), (f) cars (yellow). The 'clutter' class contains areas for which a class could not be assigned e.g. water, vertical walls, areas where computing the depth (SfM+MVS) has failed, etc. The 'low vegetation/natural ground' class contains areas on the ground covered by vegetation other than trees such as low bushes, grass, etc.

V. Network Architecture

A plethora of network architectures have already been reported for semantic labeling. State-of-the-art performance is generally associated with how deep and wide the networks are. However this introduces a significant drawback since the deeper/wider the network, the larger the number of parameters that need to be optimized during the training phase; and although training time is often overlooked it is an important factor when accessing the overall performance of a network architecture.

Furthermore, when dealing with deep network architectures the resolution of the input data deteriorates as the data progresses through to the deeper layers. This often materializes as non-smooth and noisy predictions during the final upsampling stages in the network. A common way of addressing this is to smooth the predictions using a conditional random field based post-processing approach.

In our work, we propose a distinct network architecture which uniquely combines the strength of convolutional autoencoders with feed-forward links in generating smooth predictions and reducing the number of learning parameters, with the effectiveness which cardinality-enabled residual-based building blocks have shown in improving prediction accuracy and outperforming deeper/wider network architectures with less learning parameters, to address the aforementioned limitations of existing state-of-the-art. The closest related work in terms of network architecture is with SegNet presented in [3] where the concept of feeding forward *pooling indices* from the encoders to the decoders was introduced, and the ResNeXT building blocks presented in [26] where the concept of *cardinality* was introduced and was shown that increasing cardinality was more effective than deeper/wider network architectures. Hence, to summarize, the topology of the proposed network resembles that of SegNet with the main differences that the feed forward connections are between feature maps (as opposed to pooling

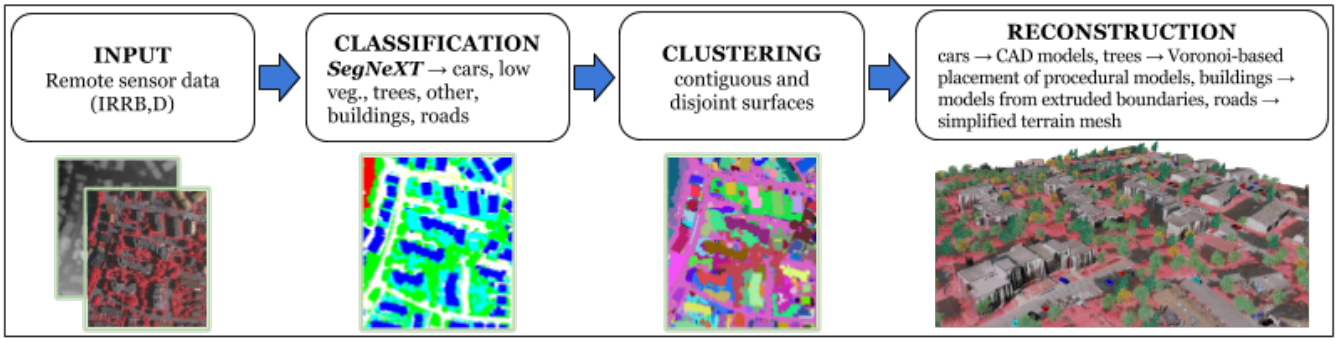


Figure 1: System Overview

indices) and each encoder or decoder consists of a ResNeXT block.

The main structure of our network can be seen in Figure 2 in comparison with SegNet and ResNeXT architectures. The network consists of a set of encoders followed by a corresponding set of decoders. Information in the form of *feature maps* - as opposed to SegNet's pooling indices - is fed forward to each decoder from its respective encoder. This results in retaining high-frequency information therefore improving boundary delineation which in turn results in smoother predictions.

The internal architecture of each encoder is that of a ResNeXT block. These blocks consist of convolutions applied across a group of feature maps that have been evenly split and are then concatenated back together. For example, a block inputs a set of 128 feature maps that are then sliced into 4 sets (i.e. cardinality is 4) of 32 feature maps, a different convolution kernel is applied to each set and the resulting feature maps are concatenated back into the 128. This operation is known as the split-transform-merge technique [26]. All other convolutions are followed by a batch normalization [11] and a ReLU activation layer.

A. Training

The network is trained on all of the available data with their corresponding ground truth. We decided to forego validation in order to maximize the available data for training. The training is performed for 2000 epochs on a single nVidia GTX 1070. We have used the Keras (with Theano) API for the development and training/testing of the network, and the code will be made available as open source. Our network trains using Keras' generator method which establishes one epoch after a certain stepsize, 64 in our case. This means each epoch consists of 64 groups containing the 32-batch samples. We trained for 11 days for 2000 epochs with each epoch requiring 500 seconds to complete.

1) Data Input: Our network takes in patches of 150×150 that are selected from the 16 available training images. We select random points within each image and construct a patch around each point. This decision is based on the observations that (a) given the high resolution of the images random sampling results in patches with large content variability, and (b) due to this large content variability (i.e. 16×5 million patches per $2K \times 2.5K$ image) the overall accuracy on the training images can be used as a proxy (i.e. almost the same) for the overall accuracy on the validation/testing images.

Each patch is represented as a 2×4-dimensional tensor containing the IRRG data and the corresponding depth map. We select 2 patches from each image-pair on each batch resulting in a total batch size of 32 different patches of $150 \times 150 \times 4$. Class balancing was also tested but did not show any improvement in learning which could be attributed to the large amount of data used.

B. Validation and Comparisons

The proposed network was validated against the additional 17 image pairs available for testing. In order to generate our test images we employ a sliding window approach that evaluates each patch at intervals of 10 pixels in the diagonal in order to minimize context based errors. Our results are then averaged into one final image.

The network's performance is measured in terms of Precision (P), Recall (R), and F_1 score which are defined as,

$$P = \frac{tp}{tp + fp} \qquad R = \frac{tp}{tp + fn} \qquad F_1 = 2 \times \frac{P \times R}{P + R} \tag{1}$$

where tp indicates the true positives, fp indicates the false positives, and fn indicates the false negatives.

Table I shows the evaluation of the overall classification results for the 17 test images and as it can be seen the overall accuracy is 89.2%. At the moment of writing the highest overall performance is 91.2% from a deep fully-convolutional neural network (FCN) ensemble followed by post-processing using a fully connected CRF (F-CRF) for further improving the results. Close inspection of our evaluation results indicate that there is about a $1 - 2\%$ variation between our classification (buildings, trees, roads, and clutter) and the state-of-the-art, and a highest difference

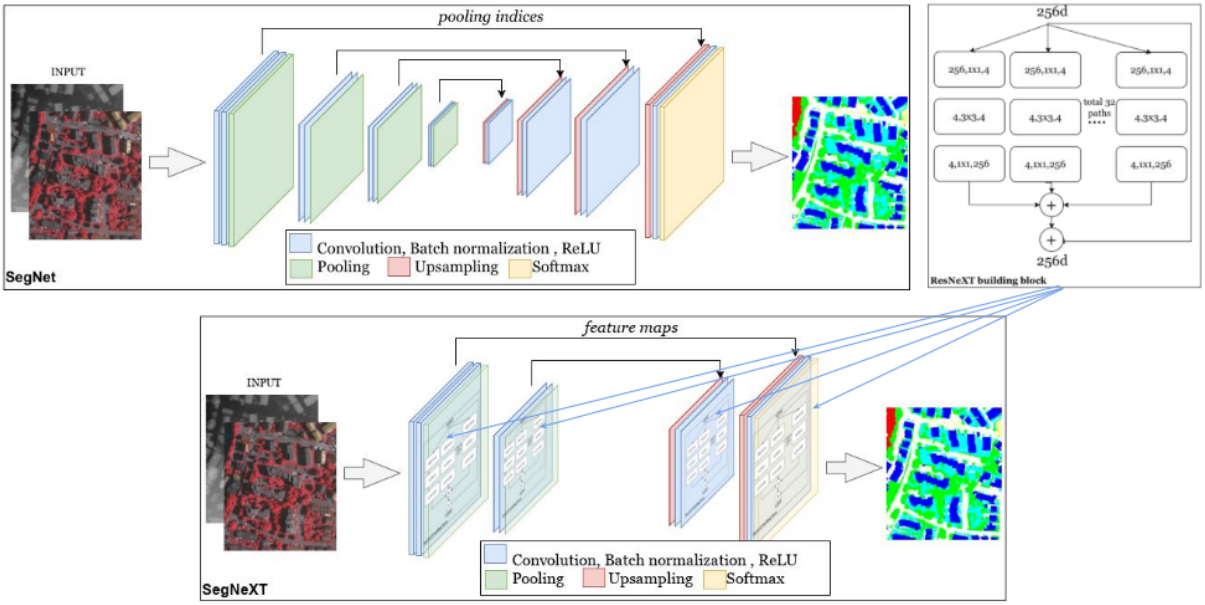


Figure 2: Top: SegNet architecture, ResNeXT block. Bottom: Our SegNeXT architecture.

of about 4% for the car class. We can only assume that the ensemble consists of networks tuned at different scales although no publication is available to corroborate this. Despite the lower overall accuracy on the entire test dataset, there are cases where the overall accuracy of our network outperforms the state-of-the-art, such as test images V-0004 shown in Figure 10b. This can be attributed to the fact that our network is under-performing (compared to state-of-the-art) in classifying cars so in the presence of many cars in the test image the overall accuracy drops; similarly, in cases where not many cars are present in the test images the overall accuracy increases and surpasses other competing networks. Figure 3 shows the results for one of the 17 test images, namely V-0003.

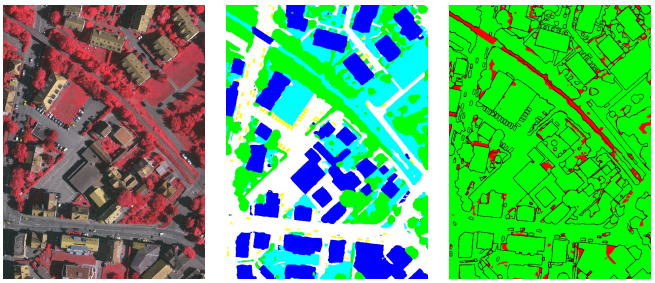
Furthermore, we have also tested variations of the proposed network architecture. In particular, we have experimented with (a) data augmentation, (b) atrous convolution [5], (c) CRF-based post-processing. There were no improvements in the overall accuracy which was in fact lower in the range of $84 - 89\%$. For the CRF-based post-processing the results were almost identical i.e. 89.1% which verifies the claim that deep autoencoders with feed-forward links between feature maps produce smooth, non-noisy results.

VI. Urban Reconstruction

The result of SegNeXT is a per-pixel classification into one of the six classes i.e. cars, buildings, trees, roads, low vegetation, and clutter. Next, the 2D per-pixel classification image is used as a proxy to cluster the 3D points in the depth map. This results in a cluster map shown in Figure 4b containing disjoint, contiguous regions. Based on the cluster's classification, an automatic per-class reconstruction is performed to generate the 3D models for each class. Finally,

↓ pred., ref. →	roads	building	low veg,	tree	car	clutter
roads	0.937	0.023	0.031	0.007	0.002	0.000
building	0.048	0.931	0.018	0.003	0.000	0.000
low veg.	0.047	0.015	0.800	0.138	0.000	0.000
tree	0.010	0.003	0.077	0.911	0.000	0.000
car	0.209	0.056	0.006	0.003	0.726	0.000
clutter	0.379	0.345	0.016	0.003	0.054	0.204
Precision	0.896	0.949	0.847	0.865	0.862	0.979
Recall	0.937	0.931	0.800	0.911	0.726	0.204
F1	0.916	0.940	0.823	0.887	0.788	0.338

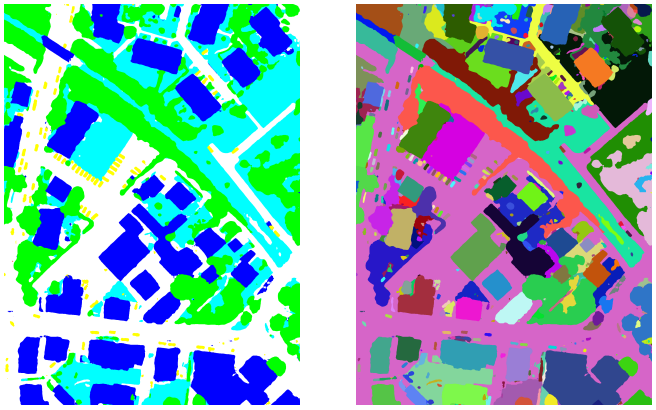
Table I: The overall evaluation of the classification results for the 17 test images for which ground truth was not provided. The network performance statistics were computed by and provided by the ISPRS Working Group II/4 organizers as part of their urban classification benchmark. All shown values are percentages.



(a) (b) (c)

Figure 3: The evaluation result for one of the 17 test images. Resolution 1887×2557 (a) Satellite image of the urban area V-0003. (b) Generated label map. (c) Red/green image, indicating wrongly classified pixels. The railroad is classified instead of clutter as a road, which is probably more close.

the 3D models are fused together to create a complete virtual representation of the entire site.



(a) (b)

Figure 4: (a) SegNeXT predictions for test image V-0003. (b) Agglomerate clustering of neighbouring points in (a) of the same class.

A. *Buildings*

Perhaps one of the most important aspects in urban reconstruction is the accurate modeling of buildings where large depth discontinuities appear as jagged edges in the captured remote sensing data. Typically, a refinement and smoothing is performed as a first step prior to further processing.

The main characteristic of the proposed SegNeXT architecture is the fact that its predictions are smooth and do not require a post-processing step of CRF-based smoothing. As it can be seen from the results SegNeXT produces smooth and non-noisy results which can be extruded to generate 3D models with smooth boundaries. The 3D points corresponding to clusters classified as buildings are triangulated using a Delaunay triangulation. Figure 5 shows examples of reconstructed buildings. Although the accuracy for buildings is in the high ninetieth percentile for all test images there are cases of misclassified clusters which cannot be identified correctly without additional information.

B. *Cars*

Clusters classified as car are removed from the geometry and replaced by simplified car CAD models. Principal Component Analysis (PCA) is performed to determine the dominant orientation of the cluster. Using the dominant orientation the CAD model is correctly placed and oriented with respect to the cluster. Cases where two cars are parked in very close proximity may lead to misclassification. An example is shown in Figure 6. To identify these cases and resolve them we examine the ratio of the cluster's eigenvalues, and its overall area measured in pixels.

From the classifications of the 16 training images we compute the average area of a car; $area_{avg}^{cars}$, and the average

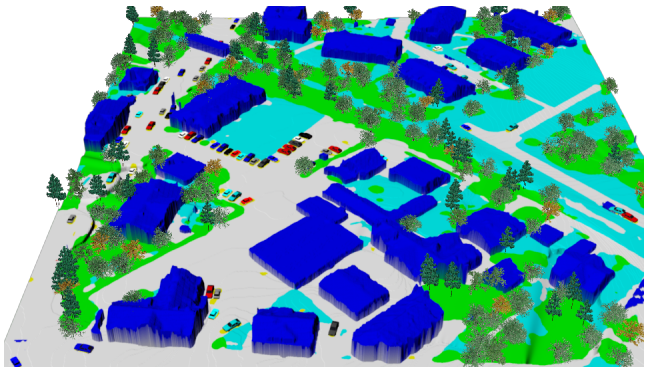


Figure 5: Buildings boundaries are extruded and roof points are triangulated. SegNeXT classification image used as texture.

ratio of the eigenvalues; $r_{avg}^{cars} = \frac{1}{N}\sum_{i=0}^{N}(\frac{\lambda_{max}}{\lambda_{min}})_{C_i}$ where $0 \leq i \leq N$ and N is the number of all car clusters C_i. Our experiments show that $area_{avg} \approx 2900$ and $r_{avg} \approx 7$ for a car in a remote sensing image with sampling density of $9cms$. Car clusters with $area^{C_i}$ significantly greater than $area_{avg}$ are further processed to determine whether they represent cars in a line ($r^{C_i} > r_{avg}$) or side-by-side ($r^{C_i} < r_{avg}$). Thus, if T cars appear in line then the car i is placed at location $i - 0.5 \times T \times (r_{avg}cos(\theta), -r_{avg}sin(\theta), 0)$ where θ is the angle formed between the dominant orientation of the current cluster's eigenvectors in terms of the world's X-Y axes. If the T cars are side-by-side the car i is placed at location $i - 0.5 \times T \times (-r_{avg}sin(\theta), r_{avg}cos(\theta), 0)$. If a cluster $(\frac{area_{avg}}{2}) < area^{C_i} < area_{avg}$ then a car is still placed to account for potential errors in classification otherwise the cluster is ignored.

To increase realism a variety of 10 CAD car models has been used. An example resolution is shown in Figure 6. Finally, since the depth map contains the depth values for the roof of the car, we use the average depth value of the hole-filled area corresponding to the car as the ground value in order to place the car. Figure 7 shows a close-up of a larger area including a parking lot where the cars have been replaced.

C. *Trees*

Clusters classified as trees are removed from the geometry and are replaced by procedurally generated models. Trees do not have a uniform shape or density which makes it impossible to identify the number of trunks and their precise location without additional information e.g. last pulse from LiDAR. In this work we address this problem by subdividing each classified tree cluster using the Voronoi tessellation. A procedurally generated tree is placed on the Voronoi center which results in a more evenly distributed placement of the trees. An example of tree clusters is shown in Figure 5 and a close up of the procedurally generated models in Figure

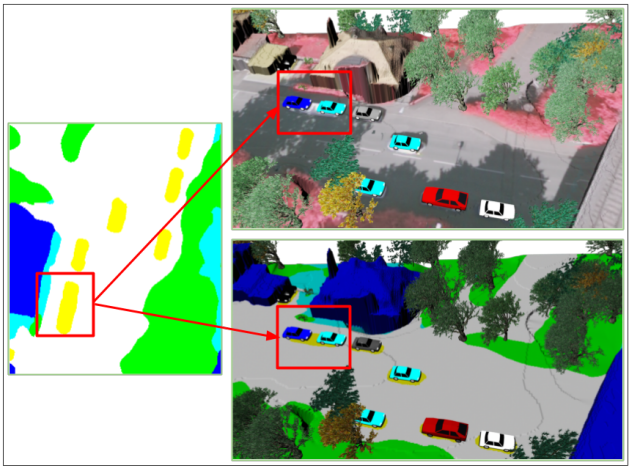


Figure 6: (left) A closeup of a SegNeXT classification result where two cars in a row are classified as one merged cluster. (right) The ratio of eigenvalues and the area are used to identify and appropriately handle special cases such as the one shown on the left where cars are parked in line and in very close proximity. This causes the classification to merge the two clusters into one.

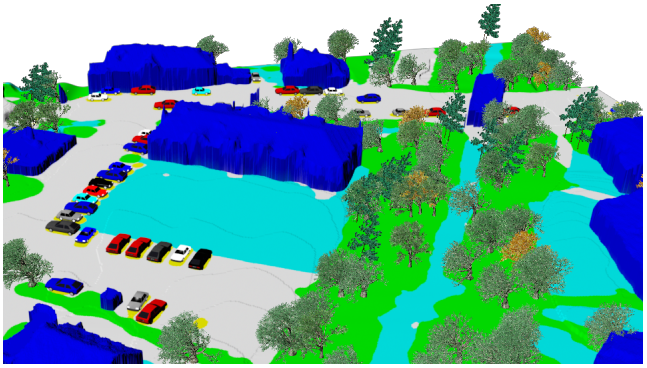


Figure 7: Close-up of a larger area which includes a parking lot demonstrating the cars' modeling and placement.

8. The Voronoi diagram for the same tree clusters is shown in Figure 9. Similarly with the cars, the depth values of a tree correspond to the top branches of the tree and not to the ground, therefore we use the depth value of the nearest classified road or low vegetation point for the placement of the tree trunk. To increase realism a variety of 8 procedurally generated models have been used.

D. Roads, Low vegetation and Clutter

Points contained in road and low vegetation are grouped together and are converted into a mesh using nearest neighbour triangulation. Holes resulting from the removal of buildings, trees, cars and clutter are filled-in using neighbourhood information. Finally, the dense mesh is further reduced by simplification. Points classified as clutter are

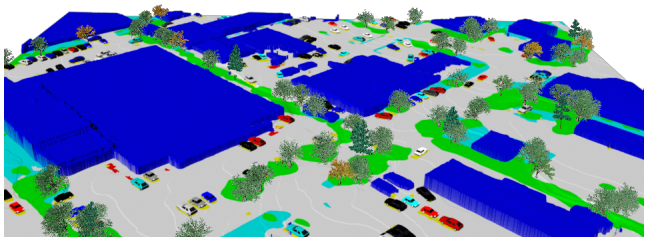
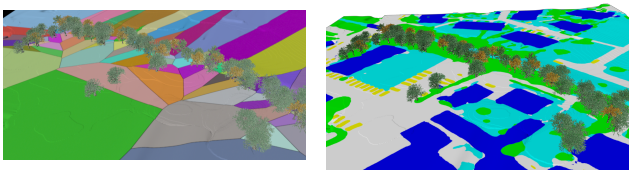


Figure 8: Classified tree clusters are removed from the geometry and are replaced by procedurally generated models.



(a) (b)

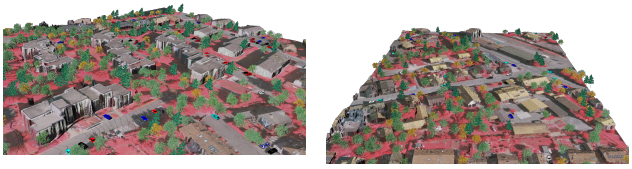
Figure 9: (a) The Voronoi diagram for the tree cluster in (b). Voronoi clusters are color-coded and centers where the tree trunks are positions are marked as black dots.

ignored as they do not represent any constant representation e.g. vertical walls, railroads, areas where SfM+MVS failed, etc.

VII. Experimental Results

We have tested the proposed framework on all 17 test images of the benchmark dataset. The average time of processing the semantic labels and converting them to 3D models is 10 minutes.

Figure 10a shows the results from V-0000. The provided orthophoto IRRG images are used to texture the models; no facade information is available. The final result of V-0004 is shown in Figure 10b. In Figure 11 we show the result for V-0013.



(a) Final result for V-0000. (b) Final result for V-0004.

VIII. Conclusion

We have presented a complete framework for urban reconstruction based on semantic labeling. Our contribution is two-fold: First, we have presented a novel network architecture which uniquely leverages the strengths of deep convolutional autoencoders with feed forward links and cardinality-enabled ResNeXT blocks. The network is shown to produce smooth results without the need for CRF-based

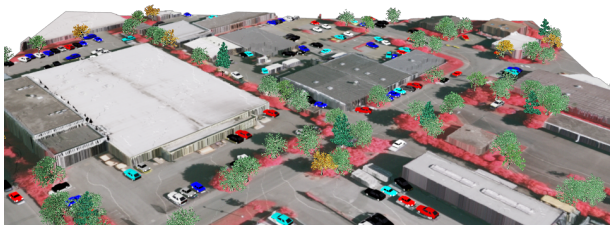


Figure 11: Final result for V-0013.

post-processing. The results on benchmark data indicate that the proposed technique can produce comparable and in some cases better classification with less computational requirements and less training time.

Secondly, we have proposed a pipeline for the automatic reconstruction of urban areas based on semantic labeling. An agglomerative clustering is performed on the points based on their class. Each cluster is further processed according to its class and generic objects such as trees and cars are removed and replaced by procedurally generated tree models and car CAD models, respectively. Buildings' boundaries are extracted, extruded and triangulated to generate 3D models. All other classes are triangulated and simplified to form a digital terrain model.

Finally, we have extensively tested the proposed framework on all 17 test images and show the realistic virtual environments generated as a result [2]. Future work includes the investigation of recently proposed architectures for improved semantic labeling such as dynamic routing capsules [22] in the deep autoencoders which have already achieved state-of-the-art performance, and the exploration of inverse solid geometry for large-scale urban reconstruction which by design resolve the problem of noisy boundaries in the reconstructed 3D models.

Acknowledgment

This research is based upon work supported by the Natural Sciences and Engineering Research Council of Canada Grants No. N01670 (Discovery Grant) and DNDPJ515556-17 (Collaborative Research and Development with the Department of National Defence Grant).

References

[1] Suhib Alsisan and Niloy J Mitra. "Variation-Factored Encoding of Facade Images." In: *Eurographics (Short Papers)*. 2012, pp. 37–40.

[2] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. "SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation". In: *arXiv:1511.00561 [cs]* (Nov. 2, 2015).

[3] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. "Segnet: A deep convolutional encoder-decoder architecture for image segmentation". In: *arXiv preprint arXiv:1511.00561* (2015).

[4] Liang-Chieh Chen et al. "DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs". In: *arXiv:1606.00915 [cs]* (June 2, 2016).

[2] Renderings and videos of the results can be downloaded from here

[5] Liang-Chieh Chen et al. "Rethinking atrous convolution for semantic image segmentation". In: *arXiv preprint arXiv:1706.05587* (2017).

[6] Liang-Chieh Chen et al. "Semantic Image Segmentation with Deep Convolutional Nets and Fully Connected CRFs". In: *arXiv:1412.7062 [cs]* (Dec. 22, 2014). arXiv: 1412.7062.

[7] Liang-Chieh Chen et al. "Semantic Image Segmentation with Task-Specific Edge Detection Using CNNs and a Discriminatively Trained Domain Transform". In: *arXiv:1511.03328 [cs]* (Nov. 10, 2015). arXiv: 1511.03328.

[8] Yanhua Cheng et al. "Locality-Sensitive Deconvolution Networks with Gated Fusion for RGB-D Indoor Semantic Segmentation". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 3029–3037.

[9] Bharath Hariharan et al. "Hypercolumns for Object Segmentation and Fine-grained Localization". In: *arXiv:1411.5752 [cs]* (Nov. 20, 2014). arXiv: 1411.5752.

[10] Yuanlie He, Sudhir Mudur, and Charalambos Poullis. "Multi-label Pixelwise Classification for Reconstruction of Large-scale Urban Areas". In: *arXiv preprint arXiv:1709.07368* (2017).

[11] Sergey Ioffe and Christian Szegedy. "Batch normalization: Accelerating deep network training by reducing internal covariate shift". In: *International Conference on Machine Learning*. 2015, pp. 448–456.

[12] Guosheng Lin et al. "RefineNet: Multi-Path Refinement Networks for High-Resolution Semantic Segmentation". In: *arXiv:1611.06612 [cs]* (Nov. 20, 2016). arXiv: 1611.06612.

[13] Wei Liu, Andrew Rabinovich, and Alexander C. Berg. "ParseNet: Looking Wider to See Better". In: *arXiv:1506.04579 [cs]* (June 15, 2015). arXiv: 1506.04579.

[14] Jonathan Long, Evan Shelhamer, and Trevor Darrell. "Fully Convolutional Networks for Semantic Segmentation". In: *arXiv:1411.4038 [cs]* (Nov. 14, 2014). arXiv: 1411.4038.

[15] Przemyslaw Musialski et al. "A survey of urban reconstruction". In: *Computer graphics forum*. Vol. 32. 6. Wiley Online Library. 2013, pp. 146–177.

[16] Hyeonwoo Noh, Seunghoon Hong, and Bohyung Han. "Learning Deconvolution Network for Semantic Segmentation". In: *arXiv:1505.04366 [cs]* (May 17, 2015). arXiv: 1505.04366.

[17] Chao Peng et al. "Large Kernel Matters – Improve Semantic Segmentation by Global Convolutional Network". In: *arXiv:1703.02719 [cs]* (Mar. 8, 2017). arXiv: 1703.02719.

[18] Tobias Pohlen et al. "Full-Resolution Residual Networks for Semantic Segmentation in Street Scenes". In: *arXiv:1611.08323 [cs]* (Nov. 24, 2016). arXiv: 1611.08323.

[19] Charalambos Poullis. "A framework for automatic modeling from point cloud data". In: *IEEE transactions on pattern analysis and machine intelligence* 35.11 (2013), pp. 2563–2575.

[20] Charalambos Poullis and Suya You. "Automatic reconstruction of cities from remote sensor data". In: *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE. 2009, pp. 2775–2782.

[21] Franz Rottensteiner et al. "Results of the ISPRS benchmark on urban object detection and 3D building reconstruction". In: *ISPRS Journal of Photogrammetry and Remote Sensing* 93 (2014), pp. 256–271.

[22] Sara Sabour, Nicholas Frosst, and Geoffrey Hinton. "Dynamic Routing between Capsules". In: 2017.

[23] Jamie Shotton et al. "TextonBoost for Image Understanding: Multi-Class Object Recognition and Segmentation by Jointly Modeling Texture, Layout, and Context". In: *Int. Journal of Computer Vision (IJCV)* (Jan. 1, 2009).

[24] Yannick Verdie, Florent Lafarge, and Pierre Alliez. *LOD generation for urban scenes*. Tech. rep. ACM, 2015.

[25] Jianxiong Xiao and Yasutaka Furukawa. "Reconstructing the worlds museums". In: *IJCV* 110.3 (2014), pp. 243–258.

[26] Saining Xie et al. "Aggregated residual transformations for deep neural networks". In: *arXiv preprint arXiv:1611.05431* (2016).

[27] Hengshuang Zhao et al. "Pyramid Scene Parsing Network". In: *arXiv:1612.01105 [cs]* (Dec. 4, 2016). arXiv: 1612.01105.

[28] Qian-Yi Zhou and Ulrich Neumann. "2.5 D building modeling by discovering global regularities". In: *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE. 2012, pp. 326–333.