Fusion of Aerial Lidar and Images for Road Segmentation with Deep CNN

Biswas Parajuli
Florida State University
Tallahassee, Florida, USA
parajuli@cs.fsu.edu

Piyush Kumar
Florida State University
Tallahassee, Florida, USA
piyush@cs.fsu.edu

Tathagata Mukherjee
Intelligent Robotics
Tallahassee, Florida, USA
tathagata@intelligentrobotics.org

Eduardo Pasiliao
Air Force Research Laboratory
Eglin AFB, Florida, USA
eduardo.pasiliao@us.af.mil

Sachin Jambawalikar
Columbia University Medical Center
New York, USA
sj2532@columbia.edu

ABSTRACT

In this paper we attempt to fuse both airborne Lidar and high resolution images (0.5 feet per pixel) to identify road networks in a large geographic region. We perform pixel-wise segmentation to classify each pixel as road or non-road based on color and depth features in a larger neighborhood context. This constitutes a bimodal setting because the RGB pixels represent the color space and the depth values come from three dimensional Lidar readings. We present multiple strategies for fusing Lidar and images. We describe a cost-effective, modular, deep convolution network design, TriSeg which gives better IoU metric for the aerial road segmentation problem than the state of the art RGB only architectures. We report on many other architectures as well as release our dataset for further research: https://bitbucket.org/biswas/fusion_lidar_images.

CCS CONCEPTS

• **Computing methodologies** → **3D imaging**; **Image segmentation**;

KEYWORDS

Convolutional Neural Network, Bimodal learning, Lidar, Road segmentation, Image processing

ACM Reference Format:
Biswas Parajuli, Piyush Kumar, Tathagata Mukherjee, Eduardo Pasiliao, and Sachin Jambawalikar. 2018. Fusion of Aerial Lidar and Images for Road Segmentation with Deep CNN. In *26th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (SIGSPATIAL '18), November 6–9, 2018, Seattle, WA, USA.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3274895.3274993

1 INTRODUCTION

The problem of identifying roads in aerial imagery arises in many applications, such as mapping, automatic road navigation and surveillance [3, 17], They all require accurate road segmentation which is mostly done manually. Further, existing applications seldom utilize both Lidar and RGB images in conjunction because they differ in their modalities, fusing them is not trivial and it is computationally expensive to process them together. In this paper, we present fully convolutional architectures which fuse the two types of data. We use a simple Lidar processing unit, along with a stack of 3 SegNets [1] to achieve better segmentation than just using the RGB images.

Road network detection in remotely sensed imagery is not a new problem. There are unsupervised methods that rely on heuristics, visual spectral properties, pre-defined shapes and edges [2, 12]. With sufficient annotated training images, classical supervised methods can also be used. We can feed per pixel neighborhood features and class labels to a learning algorithm like Support Vector Machine [14] to learn a classification model. However, this approach has a few problems: (a) non-uniform road widths affect the neighborhood size (b) high resolution images result in billions of training examples (c) individual pixels are classified independently (d) feature engineering is not trivial. To avoid these problems, Mnih et. al. [18] use a neural network which scales to large datasets well and automatically learns features from 64×64 pixel neighborhood to classify the corresponding 16×16 patches at the centers. They show that having a larger context is better for detecting roads.

More recent fully convolutional neural network (CNN) architectures for semantic segmentation, for example, FCN [16], SegNet [1] and DeepLab [6], built in the encoder-decoder framework [13] allow even larger contexts. These deep CNNs first encode the input image into coarse output maps which are then upsampled to dense pixels in the decoder stage. This gives an end-to-end, pixels-to-pixels, low memory system that maps complete input images to their corresponding complete segmentation masks. For road specific segmentation, most of the state of the art solutions apply deep networks but use only aerial images [3, 7, 17]. Our fusion architecture can be built from any of the existing network designs and only depends on the availability of resources. We use SegNet because it has a memory efficient upsampling scheme that allows larger training batch sizes.

The next challenge is to combine both images and Lidar to create spatially meaningful image-like inputs to CNNs. Since the RGB image pixels represent the color space and the depth values come from three dimensional (3D) Lidar readings, mixing the two in this bimodal setting is not trivial [11]. Moreover, Lidar scans are coarser

SIGSPATIAL '18, November 6–9, 2018, Seattle, WA, USA

ACM ISBN 978-1-4503-5889-7/18/11.
https://doi.org/10.1145/3274895.3274993

Algorithm 1 LPU

function LPUPIXEL(D, i, j) ▷
D is the set of Lidar points
(i, j) is the pixel position being computed
α−Scaling factor
β−Percentage for robust min computation
r− Neighborhood radius size
$pos(i, j)$ is the geographic location of the pixel
$Sort_z(X)$ sorts points in X based on z-coordinate
$\mu_z(X)$ returns mean of z-coordinates of points in X
$d(x, y)$ is the geographic distance between two positions

$S = Sort_z(\forall p \in D \mid d((p_x, p_y), pos(i, j)) \leq r)$
$\eta = \alpha\,(\mu_z(S) - \mu_z(S[(1-\beta)|S|, |S|]))$
return $\max(\lfloor \eta \rfloor, 255)$
end function

compared to images. This requires transforming Lidar depth into depth image while preserving the 3D road geometry. For 3D scans with on-ground viewpoint, HHA encoding is used [9, 10]. In our case, roads are mostly defined by flatness in the aerial viewpoint. Thus, we propose a simple, relative height based depth encoding scheme. We then draw from [9, 16, 19] and explore multiple ways to fuse the RGB and the encoded depth images for bimodal learning. We try early, mid-level and super-late fusion methods where the CNN itself learns to combine the bimodal features. As a different approach, we first learn separate, pixel-wise, weak classifiers for RGB and depth features that give predictions. We then concatenate these predictions to get the CNN input. Next we present the details of architectures we experimented with.

2 FUSION ARCHITECTURE

One of the commonalities of the proposed architectures is the Lidar Processing Unit (LPU). This block transforms Lidar data into a single-channeled, 8-bit depth image that is then fed into various architectures. This is an alternative to computing a lot of extra geometric features [4] or creating a more complex architecture [11]. The LPU uses Algorithm 1 to compute the output per pixel and is parameterized by three scalars - α, β, and r.

Intuitively, the LPU captures flatness of the Lidar point cloud around a given position by computing the average height of the points in the neighborhood and subtracting the minimum height. It computes a robust minimum by first sorting the points in z and then taking the average of β-percent of the points with lowest z. For optimization, we only consider the last return and discard earlier returns as non-ground points.

Our LPU algorithm runs in $O(n)$ time on uniformly distributed Lidar point clouds by using a grid based nearest neighbor search. To optimize the values of α, β, and r, we do a grid search on these values using a single depth channel as input to SegNet with a total of 300 training images with roads in them and a set of 100 testing images. This optimization gave us $\alpha = 32, \beta = 0.1$, and $r = 4$ feet. An example of the output of LPU with these parameters is shown in Figure 2. We describe our fusion architectures below:

- **SegNet**: We directly borrow it from [1]. Apart from the default Softmax loss, we experimented with Dice, IoU and XOR loss.

Table 1: Number of tiles selected for train and test from the respective bins based on the percentage of road pixels

Road %	Total tiles	Random Sample Size
0	11724	1K
1-5	1570	1K
5-10	3863	1K
10-20	5135	1K
20-30	1372	1K
30-40	547	200
40-50	217	200
50-	74	60

- **PreConv**: We build a depth convolution unit (DepthCNN) with Tree of Parzen Estimators [5]. It has two convolutional layers (8 and 4 filters) with 3×3 kernels and 1 padding. For each depth channel, it produces a 4-channel image which we concatenate with the RGB image and pass into SegNet.
- **ElePreConv**: We first add an extra zero-initialized channel to the RGB image. This 4-channel image is then added elementwise to the output of DepthCNN. It resembles a step of FuseNet [11].
- **TriSeg**: It is a union of 3 separately optimized SegNets. SegNet 1 is optimized for RGB and outputs a softmax layer with probabilities of a pixel being road. SegNet 2 takes the depth image as input and has the same output as SegNet 1. SegNet 3 takes two channel input with probabilities from SegNet 1 and 2, and outputs the classification. All three SegNets are independently optimized. See Figure 5.
- **RFNorm**: We first train cheap Random Forest classifiers for individual feature channels without considering any neighborhood. They are fast to learn and can be included in the preprocessing stage. We perform hyperparameter optimization to fix the depth and the number of trees in the Random Forest for each channel. They output per-channel, per-pixel probability scores, thereby, transforming any feature into a value between 0 and 1. We then concatenate these normalized feature channels as input to CNNs.

We chose TriSeg instead of two SegNets and an FC layer because: (a) two SegNets with an FC layer did not fit in a GTX 1080 card with 11GB memory (b) the two SegNets of TriSeg can be trained on two GPU cards in parallel (c) TriSeg is more robust to hyperparameter optimization and we did not tune the parameters at all (d) it can be created by replicating a single SegNet implementation.

3 EXPERIMENTS

We experiment with architectures defined in Section 2 with varying feature sets and loss functions. Each experiment uses a single 11 GB GeForce GTX 1080 card on an Intel i7-based workstation with 32 GB RAM. The implementation used Python and Caffe [15].

Dataset: We build our dataset with TLCGIS's [1] high resolution aerial GeoTIFF images, aerial Lidar as LAS files and the high quality, manually verified segmentation of the entire Leon county. We select 63 10K $\times$ 10K tiles that cover the city of Tallahassee and split them into 500×500 resolution tiles. This dataset comes with challenges as roads: (a) have different directions (b) lie anywhere within the frame (c) cover anywhere between 0 to greater than 50% of the

[1] http://www.tlcgis.org/

Figure 1: RGB image

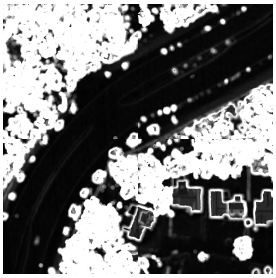


Figure 2: LPU output

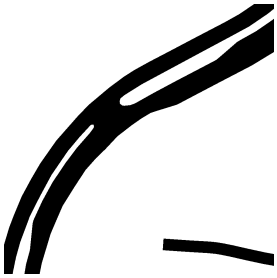


Figure 3: Ground truth

Figure 4: TriSeg output

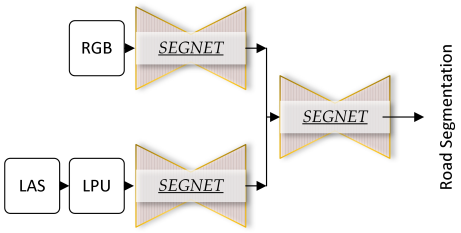
Table 2: Comparison of TriSeg with the baselines and other fusion architectures. TPR, TNR and IoU denote true positive rate, true negative rate and intersection over union. A and G stand for per test image average metric and global metric over all test images respectively.

Architecture	A-TPR	A-TNR	A-IoU	G-TPR	G-TNR	G-IoU	Iter
TriSeg	0.796	0.982	0.746	0.877	0.983	**0.784**	33K
SegNet with (RGB + Depth)	0.787	0.981	0.732	0.869	0.983	0.778	41K
RGB only SegNet	0.770	0.982	0.714	0.860	0.984	0.773	38K
DeepRoadMapper	0.748	0.985	0.692	0.839	0.986	0.763	50K
ElePreConv	0.801	0.976	0.710	0.881	0.979	0.765	31K
RFNorm on RGB and Depth	0.761	0.978	0.685	0.857	0.980	0.752	36K
HSV only SegNet	0.715	0.980	0.665	0.823	0.982	0.730	50K
Blue only SegNet	0.680	0.983	0.655	0.801	0.985	0.722	43K
PreConv	0.756	0.973	0.655	0.840	0.975	0.713	28K
Green only SegNet	0.639	0.981	0.614	0.773	0.983	0.690	35K
Red only SegNet	0.624	0.984	0.617	0.761	0.985	0.689	43K

Table 3: Effect of feature normalization with Random Forests

Input Channels for SegNet	A-TPR	A-TNR	A-IoU	G-TPR	G-TNR	G-IoU	Iter
Blue	0.680	0.983	0.655	0.801	0.985	0.722	43K
RFNorm(Blue)	0.673	0.974	0.609	0.791	0.976	0.677	38K
Depth	0.552	0.931	0.384	0.602	0.933	0.412	18K
RFNorm(Depth)	0.613	0.968	0.55	0.718	0.97	0.593	24K
RFNorm(RGB) + RFNorm(Depth)	0.761	0.978	0.685	0.857	0.980	0.752	36K
RGB + RFNorm(Depth)	0.776	0.979	0.704	0.868	0.981	0.768	40K

Figure 5: Our TriSeg architecture for processing Lidar and RGB simultaneously.



frame (Table 1). In contrast, in the CamVid dataset used in SegNet, the road is always in front of the vehicle.

Training: We first use the available road masks to bin all the tiles based on road percentage. The number of tiles picked at random from each bin is listed in Table 1. Our random sample has 5640 tiles in total. As tiles with larger road percentages are rare, we randomly select a bigger percentage of tiles from those bins. We divide the selected tiles into training, test and validation sets with each having 2640, 2400 and 240 tiles respectively.

We use the same training set to learn a different model for each variant of the architecture. We use SegNet's default hyperparameters in most cases. We fix the training mini-batch size to 4 due to the GPU memory limit. We mostly use cross-entropy (Softmax loss) as the loss function. Due to significant imbalance between road and non-road regions, we use median frequency balancing while computing the loss [8]. The median frequency is the average of the two class frequencies. We then divide the computed median frequency with the respective class frequencies to get 1.6903 and 0.2934 as weights for road and non-road respectively.

SegNet takes approximately 16 minutes for 1K iterations and 0.225 second per 512×512 input RGB image for a complete forward and backward pass. It takes similar time (15 mins) for Dice and XOR

loss functions. For one tile, our unoptimized Python implementation takes 55 seconds to compute the depth image.

Baselines and Metrics: We compare our fusion architectures with RGB-only SegNet and the DeepRoadMapper [17] road segmenter implementation provided by Bastani et. al [3]. We train the latter baseline for 50000 iterations with a batch size of 4. We use the default hyperparameters set by [3]. We also learn models for individual feature channels to evaluate improvements gained after fusion.

We use true positive rate (TPR), true negative rate (TNR) and intersection over union (IOU) as the evaluation metrics. We exclude accuracy because of the class imbalance. TPR (TNR) is the fraction of road (non-road) pixels correctly classified as roads (non-roads). IOU is an even more stringent metric. If TP, FP and FN are the true positive, false positive and false negative counts respectively, then IOU is the ratio TP/(TP + FP + FN). We further compute its per tile average measure (A-TPR, A-TNR, A-IOU) and pixelwise global measure (G-TPR, G-TNR, G-IOU). We train each model for 50K iterations and save snapshots at every 1K iteration. We use the snapshot with the best G-IOU on the validation set for testing.

Results: We report our findings in Tables 2 and 3. RGB-only-SegNet already performs almost as good as or better than most of the listed models. Contrary to [4], HSV did not help on our dataset. In the experiments with single RGB channels, we see that no individual channel is driving the full performance of RGB SegNet, although the Blue channel does the best. So, three weak features perform better when combined. Proceeding with this hypothesis, we combined depth with RGB. It did not show significant improvements. Depth alone performs poorly with only 0.41 G-IOU while direct concatenation of RGB and depth nudges the G-TPR by only 0.5%. In **PreConv** and **ElePreConv**, the DepthCNN unit learns to reduce differences in modalities between RGB and Depth. We achieve better G-TPR with **ElePreConv** but worse G-IOU indicating overflow of predicted roads. Late fusion with **TriSeg** beats RGB-only-SegNet and DeepRoadMapper by 1.3% and 2.1% respectively in G-IOU. On visual inspection we observe portions of parking lots and flat surfaces with shadows being classified as roads accounting for the spill over. There are also roads under dense canopies which the models miss. An example **TriSeg** output is given in Figure 4.

Table 3 shows that feature normalization with the RFNorm architecure is not effective when we tried to directly concatenate the input channels for fusion. The global IoU metric worsened with this approach for and Depth channels. In case of individual channels, RFNorm degraded the metric for Blue channel by 4.5% whereas it improved for Depth channel by almost 18%.

We also experimented with multiple loss functions, and their combinations, including Dice, IoU, XOR, and Softmax. IoU did not converge in $50K$ iterations. Dice seemed easier to converge than XOR and beat XOR in the first $50K$ iterations. Both Dice and XOR were overfitting, and hence we had to optimize the weight decay parameter using the L_2 norm. Dice, with 78% global IoU, came close to **TriSeg** but could not beat it.

4 CONCLUSION

We present a simple, deep convolutional network architecture that processes both aerial images and Lidar data for road segmentation. We also use a new and large dataset from our city, which has high quality imagery, Lidar and manually verified segmentation for training. Our work is based on SegNet and we compare various approaches for processing this bimodal dataset. In the future, we plan to extend our work to do aerial segmentation with more types of objects, such as buildings, water, forest, pavements and railtracks.

REFERENCES

[1] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. 2017. SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2017).

[2] Ruzena Bajcsy and Mohamad Tavakoli. 1976. Computer recognition of roads from satellite pictures. *IEEE Transactions on Systems, Man, and Cybernetics* 9 (1976), 623–637.

[3] Favyen Bastani, Songtao He, Sofiane Abbar, Mohammad Alizadeh, Hari Balakrishnan, Sanjay Chawla, Sam Madden, and David DeWitt. 2018. RoadTracer: Automatic Extraction of Road Networks from Aerial Images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 4720–4728.

[4] Carlos Becker, Nicolai Häni, Elena Rosinskaya, Emmanuel d'Angelo, and Christoph Strecha. 2017. Classification of aerial photogrammetric 3D point clouds. *arXiv preprint arXiv:1705.08374* (2017).

[5] James Bergstra, Daniel Yamins, and David Cox. 2013. Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures. In *International Conference on Machine Learning*. 115–123.

[6] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. 2018. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence* 40, 4 (2018), 834–848.

[7] Guangliang Cheng, Ying Wang, Shibiao Xu, Hongzhen Wang, Shiming Xiang, and Chunhong Pan. 2017. Automatic road detection and centerline extraction via cascaded end-to-end convolutional neural network. *IEEE Transactions on Geoscience and Remote Sensing* 55, 6 (2017), 3322–3337.

[8] David Eigen and Rob Fergus. 2014. Predicting Depth, Surface Normals and Semantic Labels with a Common Multi-Scale Convolutional Architecture. *CoRR* abs/1411.4734 (2014). arXiv:1411.4734 http://arxiv.org/abs/1411.4734

[9] Andreas Eitel, Jost Tobias Springenberg, Luciano Spinello, Martin Riedmiller, and Wolfram Burgard. 2015. Multimodal deep learning for robust rgb-d object recognition. In *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*. IEEE, 681–687.

[10] Saurabh Gupta, Pablo Arbelaez, and Jitendra Malik. 2013. Perceptual organization and recognition of indoor scenes from RGB-D images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 564–571.

[11] Caner Hazirbas, Lingni Ma, Csaba Domokos, and Daniel Cremers. 2016. Fusenet: Incorporating depth into semantic segmentation via fusion-based cnn architecture. In *Asian Conference on Computer Vision*. Springer, 213–228.

[12] Xiangyun Hu, C Vincent Tao, and Yong Hu. 2004. Automatic road extraction from dense urban area by integrated processing of high resolution imagery and lidar data. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences. Istanbul, Turkey* 35 (2004), B3.

[13] Fu Jie Huang, Y-Lan Boureau, Yann LeCun, et al. 2007. Unsupervised learning of invariant feature hierarchies with applications to object recognition. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*. IEEE, 1–8.

[14] Xin Huang and Liangpei Zhang. 2009. Road centreline extraction from high-resolution imagery based on multiscale structural features and support vector machines. *International Journal of Remote Sensing* 30, 8 (2009), 1977–1987.

[15] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. 2014. Caffe: Convolutional Architecture for Fast Feature Embedding. *arXiv preprint arXiv:1408.5093* (2014).

[16] Jonathan Long, Evan Shelhamer, and Trevor Darrell. 2015. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 3431–3440.

[17] Gellért Máttyus, Wenjie Luo, and Raquel Urtasun. 2017. Deeproadmapper: Extracting road topology from aerial images. In *International Conference on Computer Vision*, Vol. 2.

[18] Volodymyr Mnih and Geoffrey Hinton. 2010. Learning to detect roads in high-resolution aerial images. *Computer Vision–ECCV 2010* (2010), 210–223.

[19] Jiquan Ngiam, Aditya Khosla, Mingyu Kim, Juhan Nam, Honglak Lee, and Andrew Y Ng. 2011. Multimodal deep learning. In *Proceedings of the 28th international conference on machine learning (ICML-11)*. 689–696.