Combining Satellite Imagery and GPS Data for Road Extraction

Tao Sun, Zonglin Di, Yin Wang
Tongji University, Shanghai, China
{suntao,1452640,yinw}@tongji.edu.cn

ABSTRACT

Road extraction is a fundamental problem in remote sensing and mapping. Recent advances in Convolution Neural Network (CNN) have contributed significant improvements in automatic road extraction from satellite imagery, albeit prediction gaps challenge post-processing. Some of the gaps are hard to bridge by satellite imagery alone due to dense vegetation, road construction, and building shadows. In this paper, we combine satellite imagery with GPS data to improve road extraction quality. Our dataset includes 100cm pixel resolution satellite imagery and 192-hour taxi GPS traces from the urban area of Beijing. Experimenting with various layers to combine GPS data, our CNN model outperforms the RGB-only model by nearly 13% on mean IoU.

CCS CONCEPTS

• **Computing methodologies** → **Image segmentation**; • **Information systems** → *Geographic information systems*;

KEYWORDS

Road extraction, GPS, Satellite image, U-Net

ACM Reference Format:
Tao Sun, Zonglin Di, Yin Wang. 2018. Combining Satellite Imagery and GPS Data for Road Extraction. In *2nd ACM SIGSPATIAL International Workshop on AI for Geographic Knowledge Discovery (GeoAI'18), November 6, 2018, Seattle, WA, USA.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3281548.3281550

1 INTRODUCTION

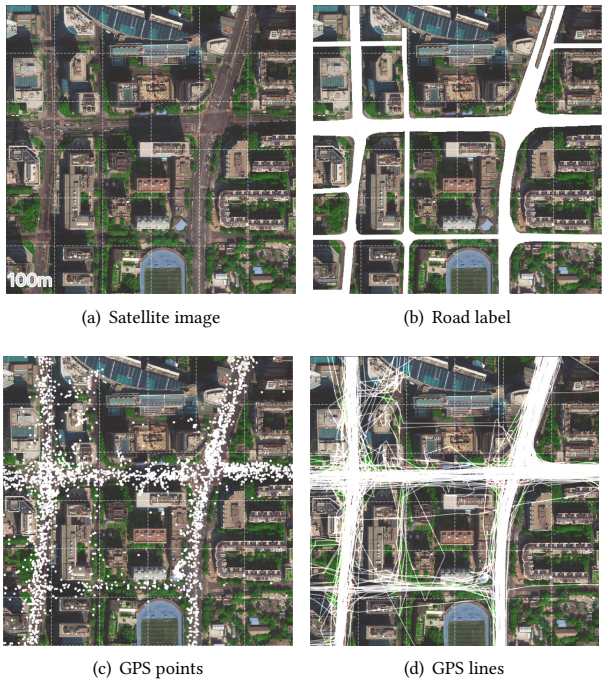
Our modern lives heavily rely on accurate road maps, and future autonomous driving would demand higher precision and more up-to-date maps. The traditional mapping solution is labor intensive that requires land survey, manual editing, and verification. Manually created maps are costly, error-prone, and easily become outdated.

Recent advances in Convolutional Neural Networks (CNN) have inspired automatic road extraction using satellite imagery [1–5]. The improvement of CNN over traditional machine learning solutions is as magical as CNN in other typical computer vision tasks such as image classification. However, a small percentage of prediction errors pose big challenges for automated or semi-automated mapping, because these errors lead to road gaps and spurious roads that are absolutely unacceptable for maps. Finding and correcting these errors remain a tedious manual effort. Even though CNN technologies improve rapidly, we do not expect these errors to disappear completely because many road segments cannot be extracted from satellite imagery alone, such as dense vegetation, building shadow, and dirt roads without clear curbs.

On the other hand, GPS data is increasingly abundant and has long been proposed for map inference and map update [6, 7]. In practice, however, public GPS data typically covers only major roads, and cannot build complete road maps. Extracting accurate centerlines from noisy GPS data is also a challenge, especially with long sampling intervals and around intersections or sharp turns.

Satellite imagery and GPS data complement each other for the task of road extraction. GPS data can fill in the prediction gaps of major roads, and satellite imagery can cover less frequently traveled residential roads. The latter is relatively short and dense such that prediction gaps are less serious than those in major roads. Our study shows that commercial maps have many missing roads in residential areas as well [2]. In addition, satellite imagery can help obtain accurate centerlines since public GPS data is often noisy and sparse.

In this paper, we propose to combine satellite imagery and GPS data for road extraction. Our dataset includes satellite imagery and 192-hour taxi GPS data from Beijing. Our CNN model is based on U-Net [8], a popular choice in the recent DeepGlobe challenge [5].



(a) Satellite image (b) Road label

(c) GPS points (d) GPS lines

Figure 1: An image patch of our dataset

GeoAI'18, November 6, 2018, Seattle, WA, USA

ACM ISBN 978-1-4503-6036-4/18/11…$15.00
https://doi.org/10.1145/3281548.3281550

We experiment with different ways to incorporate GPS data into our model, and compare the results among them, as well as the results using satellite imagery or GPS data alone.

The remainder of this paper is as follows. Section 2 presents our U-Net model and different methods to incorporate GPS data. Section 3 describes the dataset and implementation details. Section 4 shows the results and Section 5 concludes the paper.

2 OUR METHOD

This section describes our CNN model for satellite imagery and different ways to incorporate GPS data into the model.

2.1 Our CNN Model

Figure 1(a) and Figure 1(b) show a satellite image patch and the road pixel label (white) on top of it, respectively. Based on the image and road label, there are two categories of CNN models for road extraction from satellite imagery, image classification and image segmentation. The classification models, such as AlexNet [9], VGG [10], and ResNet [11], try to label each pixel of the center part, e.g., 32x32 pixel^2, of an image patch, e.g., 128x128 pixel^2, as either road or non-road [1]. We label the entire image by rolling windows with overlapping regions. The segmentation models, such as Full Convolutional Network (FCN) [12] and U-Net [8], try to label all pixels of the entire image patch at once. A recent road extraction competition shows that U-Net is a popular choice, and outperforms many other models [5].

More specifically, U-Net is an end-to-end semantic segmentation network built upon FCN, where more feature channels are added to the up-sampling part to allow the network to propagate the contextual information to higher resolution layers, which improves the prediction results and speeds up the training process. For our purpose, we add additional input channels the input channel from three (R, G, B) to five (R, G, B, GPS-Point, GPS-Line) to adjust the input of satellite image and GPS data. Figure 2 shows the overall structure of our U-Net.

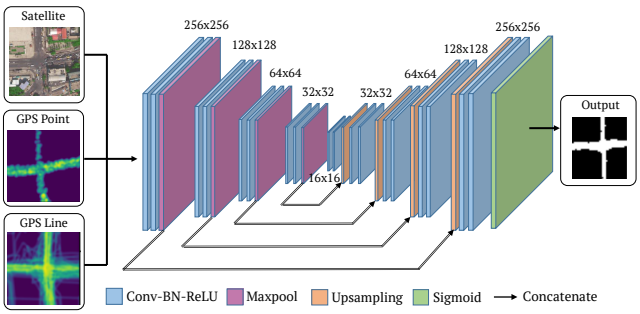


Figure 2: Pipeline of modified U-Net

2.2 GPS input channels

Since CNN models take bitmap channels, we need to rasterize GPS data. Here we adopt the Kernel Density Estimation (KDE) methods that demonstrated good results for road inferencing [6].

More specifically, there are two ways to rasterize GPS data, by points and by line segments. Figure 1(c) and Figure 1(d) show raw GPS points and line segments between consecutive GPS points, respectively. The GPS point KDE method adds a 2D Gaussian kernel for each GPS point pixel, and the GPS line KDE method adds a 2D Gaussian kernel for each pixel along each line segment. Our previous study shows that these two methods complement each other [6]. GPS points are more accurate, without any distortion along curves or intersections, but require a significant amount of data to cover the entire road, especially for less frequently traveled ones. GPS lines give good coverage even with a small amount of data, but are noisier, especially along curves and intersections, which get worse as the sampling interval increases.

While the KDE-based road inferencing methods have to use either GPS points or lines as the input, CNN models are much more flexible because they can take multiple input channels without changing any network structure. Therefore, we experiment with RGB-only, RGB+Point, RGB+Line, RGB+Point+Line, and Point+Line (without RGB) in this paper, more details below.

3 IMPLEMENTATION

We discuss our dataset and the implementation details in this section.

3.1 Dataset

We collected 120 satellite images from urban Beijing, within the Fifth Ring Road. The size of each RGB image is 1024×1024 with resolution of 100 cm per pixel. With OpenStreetMap (OSM) as reference, we labeled all paved roads manually as ground truth. For better prediction results, we manually label all pavement pixels instead of rendering a road mask image using a fixed width for each road type, see Fig. 1(b). This is consistent with previous studies [2, 5]. The road pixel coverage of our satellite images is 13.1 %.

Our GPS data is from 65 taxis in Beijing, a total 192 hours of driving. The sampling interval is 10 seconds, relatively short for taxi GPS data or other public dataset. The dataset covers most major roads within the Fifth Ring Road well. The spatial resolution of the GPS data is 0.00001 degree latitude and longitude (about 1m at Beijing).

Data disparity in GPS data is very common and our dataset is no exception [13]. Figure 3 shows the density distribution. Highways areas have two orders of magnitude more data than residential areas. We use log scale when rasterizing our GPS data using KDE methods.

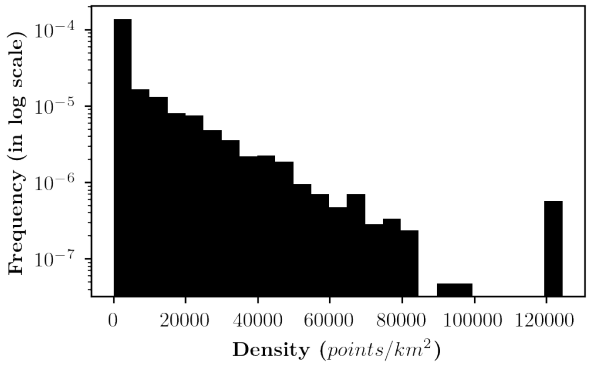


Figure 3: Density distribution of GPS points

We adopt GPS data cleaning rules in [6], removing stationary points (waiting for passengers), long sampling interval line segments, sharp turns, excessive speed, and etc.

3.2 Training Details

We train our model in PyTorch on 2 Nvidia GeForece GTX 1080Ti GPUs. As usual, we apply 5-fold cross validation and data enhancement including rotation, flipping, cropping, and HSV tuning. It takes about 50 hours for the model to converge.

We use overlapping rolling windows to smooth out the prediction images. Every input patch is 256×256 and the stride is 48. The batch size is 48. In the training process, the initial learning rate is 0.001 and we run 80 epochs. Because roads cover a small percentage of pixels, the common cross-entropy loss will result in slow convergence and bad performance. We use the Jaccard loss function to handle the uneven ratio of positive and negative pixels as

$$L = (1 - \lambda)L_{ce} - \lambda \log(L_j) \tag{1}$$

L_{ce} and L_j are defined as:

$$L_{ce} = -\frac{1}{N} \sum N_{i=0}(y \log y' + (1 - y)\log(1 - y')) \tag{2}$$

$$L_j = \frac{1}{N} \sum N_{i=0} \frac{y_i y'_i}{y_i + y'_i - y_i y'_i} \tag{3}$$

where y and y' are the target and prediction vectors, respectively, and λ is the weight of Jaccard loss, set to 0.8 [14, 15].

If the loss does not decrease over 4 epochs, we decrease the learning rate to 1/5 in the next epoch. We terminate training when the loss stops decreasing over 10 epochs or the learning rate is smaller than 1e-7.

For evaluation, we use mean intersection over union (mIoU), precision, recall and F1-measure as the metric [16]. For CNN models, we always use five input channels representing R, G, B, GPS point, and GPS line, respectively. We set the corresponding channels to zero if they are not used in an input combination. We also compare against KDE point and KDE line models described in [6] for reference. In all these cases, we choose the thresholds that have the best mIoU values in training for evaluation on test data.

4 EXPERIMENTAL RESULTS

Table 1 is our experimental results. Overall, the CNN model with both RGB and GPS input performs significantly better over models with RGB or GPS alone, especially in mIoU and F1-measure. When incorporating GPS data, using both points and lines performs marginally better over models using either of them. This is different from the KDE methods that show bigger differences using GPS points or lines, suggesting that CNN models are less sensitive to the visual differences in GPS data rendering style. Models using GPS data alone tend to have better precision, with KDE methods being the best, suggesting that the large amount of GPS data can well offset the noise we see in individual GPS traces. We plan to more thoroughly evaluate the impact of GPS noise and sparsity in road extraction in the future.

Our mIoU values using RGB only are much smaller comparing to the top results in the Deepglobe challenge [5]. This is mostly because we have a much smaller training data size, since we have to label roads manually ourselves. We could not use the DeepGlobe satellite imagery because we do not have GPS data in those regions. Although extremely encouraging to see that GPS data can significantly compensate the lack of training data, which is country specific and hard to develop, we plan to build more training data in the future and see if GPS data could still improve the results by similar percentage at a higher baseline.

Table 1: Results of mIoU, Recall, Precision and F1-Measure (F1) on the dataset after 5-fold cross-validation with KDE method as baseline

Method	Input	mIoU	Recall	Precision	F1
CNN	RGB	31.36	38.73	39.24	38.98
	RGB+Line	43.45	60.92	51.51	55.82
	RGB+Point	43.79	64.12	50.59	56.56
	RGB+Line+Point	**44.02**	**64.96**	50.48	**56.81**
	Line+Point	36.08	47.46	54.03	50.53
KDE	Line-based	31.64	39.42	**61.61**	48.08
	Point-based	34.06	46.27	56.34	50.82

Figure 4 visualizes the road extraction results of different models in four areas. In these areas, CNN model using RGB input only performs the worst, especially in precision along railways, walk paths between houses, and narrow buildings visually similar to roads. We believe more training data can improve the RGB-only model significantly. RGB-only model also has many gaps in road prediction especially along building shades and tree-covered areas, as discussed in Section 1, while models using GPS data have much more smooth predictions, and CNN models using both RGB and GPS data are the best.

5 CONCLUSION

In this paper, we proposed to combine satellite imagery and GPS data in CNN models for road extraction and mapping Our experiments using a U-Net model and our Beijing dataset show significant improvement in all evaluation metrics. The overall mIoU improvement is more than 10% compared to the CNN model using satellite imagery alone or the KDE method using GPS data alone. In addition, GPS data is much more uniform than RGB images that vary country by country and terrain by terrain, therefore greatly reduces the workload to develop region-specific RGB training data for CNN models. With the encouraging initial results, we plan to more thoroughly evaluate the effect of GPS data in CNN models in the future, including different CNN structure, different amount of training data, and GPS noise and sparsity variations.

REFERENCES

[1] Volodymyr Mnih. *Machine Learning for Aerial Image Labeling.* PhD thesis, University of Toronto, 2013.
[2] Yin Wang. Scaling Maps at Facebook. In *SIGSPATIAL*, 2016. keynote.
[3] Gellért Máttyus, Wenjie Luo, and Raquel Urtasun. Deeproadmapper: Extracting road topology from aerial images. In *The IEEE International Conference on Computer Vision (ICCV)*, 2017.
[4] Dragos Costea, Alina Marcu, Marius Leordeanu, and Emil Slusanschi. Creating roadmaps in aerial images with generative adversarial networks and smoothing-based optimization. In *ICCV Workshops*, pages 2100–2109, 2017.
[5] Ilke Demir, Krzysztof Koperski, David Lindenbaum, Guan Pang, Jing Huang, Saikat Basu, Forest Hughes, Devis Tuia, and Ramesh Raskar. Deepglobe 2018: A challenge to parse the earth through satellite images. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2018.

Figure 4: Results of four example image patches. The first column is the CNN model with RGB input alone, the second and third are KDE models using GPS lines and GPS points, and the last is the CNN model with input combination of RGB, GPS lines and GPS points. True positive, false positive, and false negative pixels are colored in green, red, and blue, respectively.

[6] Xuemei Liu, James Biagioni, Jakob Eriksson, Yin Wang, George Forman, and Yanmin Zhu. Mining large-scale, sparse gps traces for map inference: comparison of approaches. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 669–677. ACM, 2012.

[7] Yin Wang, Xuemei Liu, Hong Wei, George Forman, Chao Chen, and Yanmin Zhu. Crowdatlas: Self-updating maps for cloud and personal use. In *Proceeding of the 11th Annual International Conference on Mobile Systems, Applications, and Services*, MobiSys '13, 2013.

[8] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[9] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. ImageNet Classification with Deep Convolutional Neural Networks. *NIPS*, 2012.

[10] Karen Simonyan and Andrew Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. *CoRR*, 2014.

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[12] E. Shelhamer, J. Long, and T. Darrell. Fully convolutional networks for semantic segmentation. *IEEE Trans. on PAMI*, 39(4):640–651, April 2017.

[13] James Biagioni and Jakob Eriksson. Map inference in the face of noise and disparity. SIGSPATIAL, 2012.

[14] Tao Sun, Zehui Chen, Wenxiang Yang, and Yin Wang. Stacked u-nets with multi-output for road extraction. In *DeepGlobe*, 2018. CVPR workshop.

[15] Maxim Berman Amal Rannen Triki Matthew and B Blaschko. The lovász-softmax loss: A tractable surrogate for the optimization of the intersection-over-union measure in neural networks. 2018.

[16] Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. *Deep learning*, volume 1. MIT press Cambridge, 2016.