Leveraging Crowdsourced GPS Data for Road Extraction from Aerial Imagery

Tao Sun, Zonglin Di, Pengyu Che, Chun Liu, Yin Wang
Tongji University, Shanghai, China
{suntao, dizonglin, chepengyu, liuchun, yinw}@tongji.edu.cn

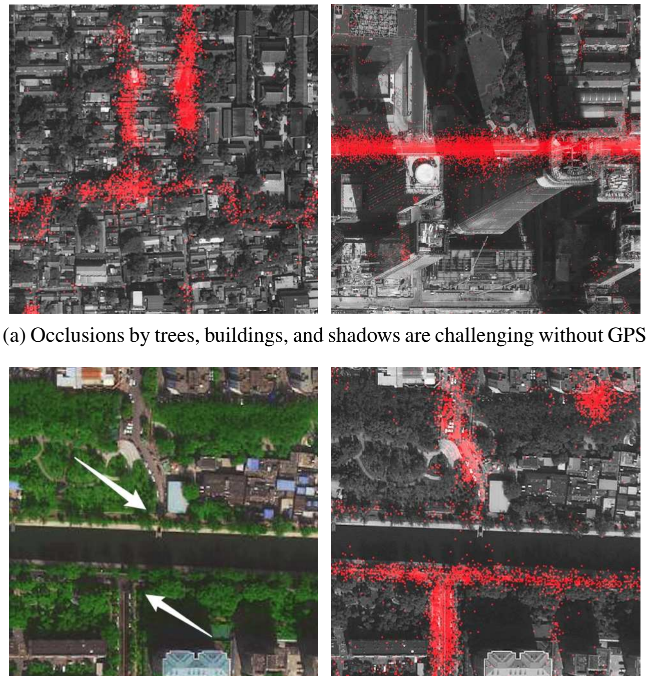
Abstract

Deep learning is revolutionizing the mapping industry. Under lightweight human curation, computer has generated almost half of the roads in Thailand on OpenStreetMap (OSM) using high resolution aerial imagery. Bing maps are displaying 125 million computer generated building polygons in the U.S. While tremendously more efficient than manual mapping, one cannot map out everything from the air. Especially for roads, a small prediction gap by image occlusion renders the entire road useless for routing. Misconnections can be more dangerous. Therefore computer based mapping often requires local verifications, which is still labor intensive. In this paper, we propose to leverage crowd sourced GPS data to improve and support road extraction from aerial imagery. Through novel data augmentation, GPS rendering, and 1D transpose convolution techniques, we show almost 5% improvements over previous competition winning models, and much better robustness when predicting new areas without any new training data or domain adaptation.

1. Introduction

Segmentation of aerial imagery has been an active research area for more than two decades [4, 17]. It is also one of the earliest applications of deep convolutional neural nets (CNN) [19]. Today, using deep convolutional neural nets over high resolution satellite imagery, Facebook has added 370 thousand km of computer generated roads to OpenStreetMap (OSM) [3] Thailand, accounting for 46 % of the total roads in the country, which is on display for all Facebook users [1, 22]. Microsoft used similar techniques to add 125 million building polygons to Bing maps U.S., five times more than those on OSM [29].

Despite real-world applications, mapping by aerial imagery has its limitations. The top challenge is overfitting. The deep neural net models often deteriorate miserably with new terrain, new building styles, new image styles, or new resolutions. Other than the model limitation, occlusions by vegetation, buildings, and shadows can be excessive. Many features are indistinguishable from the air, e.g., dirt roads and bare fields, cement pavements and building tops, alleys in slum areas. Bad weather, low satellite angle, and low light angle further complicate the issue. Even if the feature is perfectly clear, mapping often needs local knowledge. Trails and roads may have same appearances. Houses and storage sheds may have similar sizes and roofs. To make things worse, mapping has low tolerance for errors. Especially for roads, incorrect routes cause longer travel time, lead people to restricted areas, and even cause fatal accidents [34]. Because of these reasons, OSM prefers local mappers for each area, and even requires local verification for large-scale edits [1].



(a) Occlusions by trees, buildings, and shadows are challenging without GPS

(b) Roads susceptible to over connection in post-processing without GPS

Figure 1: Crowdsourced GPS data helps road extraction when aerial imagery alone is insufficient or challenging. Here each red dot represents a taxi GPS sample.

With a smart phone or any other GPS device, one can easily travel a street and verify its existence with the recorded trace. Going through all streets systematically and regularly for updates, however, is a labor intensive job that is costly and error prone. On the other hand, crowdsourced GPS data are much cheaper and increasingly abundant [6, 12, 15, 26, 34]. Figure 1 illustrates how crowdsourced GPS data, albeit noisy, can help discover roads, confirm road continuity, and avoid misconnection.

In this paper, we propose to fuse crowdsourced GPS data with aerial imagery for road extraction. Through large taxi and bus GPS datasets from Beijing and Shanghai, we show that crowdsourced GPS data has excessive noise in both variation and bias, and high degrees of disparity in density, resolution, and distribution. By rendering the GPS data as new input layers along with RGB channels in the segmentation network, together with our novel GPS data augmentation techniques and 1D transpose convolution, our model significantly outperforms existing models using images or GPS data alone. Our data augmentation is especially effective against overfitting. When predicting a new area, the performance drop is much less than the model with image input only, despite completely different GPS data quantity and resolution. We have published our code[1] and our data is available upon request.

2. Related Work

Aerial imagery segmentation has been a very active research area for a long time. We refer readers to some performance studies and references therein for early algorithms [4, 17]. Like many other image processing problems, these early solutions are often limited in accuracy and difficult to generalize to real-world datasets.

Mnih first used a deep convolutional neural net similar to LeNet [13] to extract roads and buildings from a 1.2 m/pixel dataset in the U.S. [18, 19]. Moving to developing countries with more diversified roads and buildings, Facebook showed that deeper neural nets perform much better on a 50 cm/pixel dataset [33]. Both of these early approaches convert the semantic segmentation problem into a classification problem by classifying each pixel of a center square, e.g., 32 x 32, as road or building from a larger image patch, e.g., 128 x 128. Stitching these center squares together is the final output for a large input image. Performance issues aside, this classification approach cannot learn complicated structures such as street blocks and building blocks due to limited input size.

With the commercial availability of 30 cm/pixel satellite imagery and low-cost aerial photography drones, more public high-resolution datasets become available [10, 11, 32, 35, 36]. These new datasets and industrial interests lead to a proliferation of research activities recently [5, 16, 31, 40]. Semantic segmentation models based on the fully convolutional neural net architecture become main stream [27]. In a recent challenge [10], all top solutions used variants of U-net [24] or Deeplab [8] to segment an entire image at once, up to 1024 x 1024 pixels. A larger input size gives more context, which often leads to more structured and accurate prediction results.

With new models and multi-country scale datasets, many real-world applications emerge. Most notably, Facebook has recently added 370 thousand km of roads extracted from satellite imagery to OSM [3] Thailand, or 46 % of the total roads in the country [1, 22]. Microsoft is displaying 125 million computer generated building polygons on Bing US maps, in contrast to the 23 million polygons from OSM also on display that are mostly manually created or imported [29].

Comparing to other computer vision applications, road mapping has little margin for error. Prediction gaps make the entire road useless for routing, and therefore have attracted lots of attention. Mnih noticed the problem early on and used Conditional Random Fields in post-processing to link broken roads [18]. Another popular technique to link roads is shortest path search [16, 33]. Line Integral Convolution can smooth out broken roads in post-processing too [14]. More recent works try to address the problem in prediction instead of post-processing, e.g., through a topology-aware loss function [20] or through an iterative search process guided by CNNs [5]. We must be careful to link roads because incorrect connections are more dangerous than missing connections in routing. Our approach complements the above mentioned methods because GPS data can confirm the connectivity or the absence of it regardless of image occlusion or other issues.

Road inferencing from GPS traces has been studied for a long time too [9, 23, 25]. Most early works use dense GPS samples from controlled experiments. Recent works explored crowdsourced GPS data under various sampling interval and noise levels [6, 12, 15, 26, 34]. Kernel Density Estimation is a popular method robust against GPS noise and disparity [6, 9, 15].

There is limited research work using both GPS data and aerial imagery. One idea filters out GPS noise by road segmentation before road inferencing [37]. Our preliminary work explored the idea of rendering GPS data as a new CNN input layer, but the segmentation model used was a bit outdated and the GPS data was from a controlled experiment [28]. This paper experiments with many state-of-the-art segmentation models and crowdsourced GPS datasets several orders of magnitude bigger and noisier.

3. Crowdsourced GPS Data

We collected two taxi and bus GPS datasets from Beijing and Shanghai, respectively. The Beijing dataset is about one

[1] https://github.com/suniique/

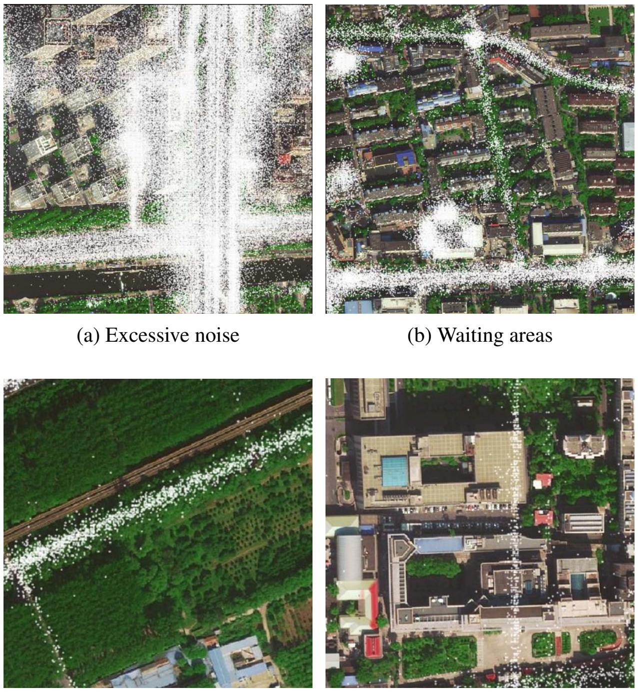
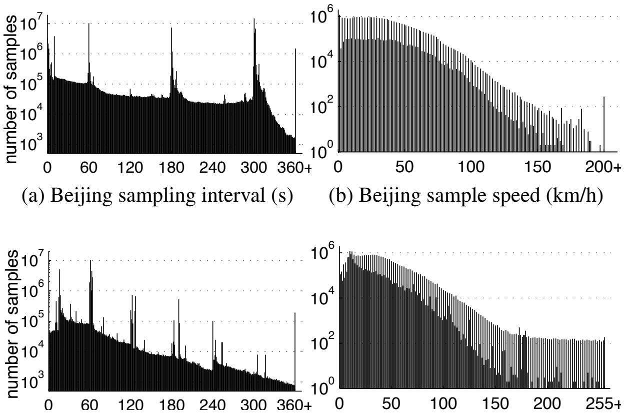

(a) Excessive noise
(b) Waiting areas
(c) Misalignment
(d) Outdated data

Figure 2: Typical issues with crowdsourced GPS data

week of data with around 28 thousand taxis and 81 million samples. The Shanghai dataset spans about half an year with around 16 thousand taxis and 1.5 billion samples. In both cases, each sample includes a timestamp, latitude, longitude, speed, bearing, and taxi status flags. Although taxis have different behaviors and trajectories than other GPS data sources, we believe many characteristics and issues in our datasets are quite representative. Therefore our method applies to other datasets.

Under ideal conditions, GPS samples follow a 2D Gaussian distribution [30]. Predicting roads can be straightforward if the samples are dense and evenly distributed. In practice, multipath errors occur in urban canyons, inside tunnels, and under elevated highways or bridges. GPS receivers vary in quality and resolution, and may integrate Kalman filters that are not Gaussian. Some datasets purposefully reduce resolution and/or add random noise for privacy protection. Figure 2a is an example of noisy GPS samples mainly due to urban canyon and elevated roads.

Even if the samples are perfectly Gaussian distributed, unlike controlled experiments or surveys, crowdsourced GPS data are not evenly distributed along each road. Highways and intersections can have orders of magnitude more data than other road areas. Some residential roads are not traveled at all. Depending on the source of data, there may be concentrations of samples in non-road areas. Figure 2b shows three high density clusters outside of artery roads, likely popular taxi waiting areas. Misalignment can occur with shifted data, e.g., Fig. 2c, or with different time periods when the data are taken, e.g., Fig. 2d.



(a) Beijing sampling interval (s)
(b) Beijing sample speed (km/h)
(c) Shanghai sampling interval (s)
(d) Shanghai sample speed (km/h)

Figure 3: Distributions of sampling interval and speed

Table 1: Typical measurement resolutions in our datasets

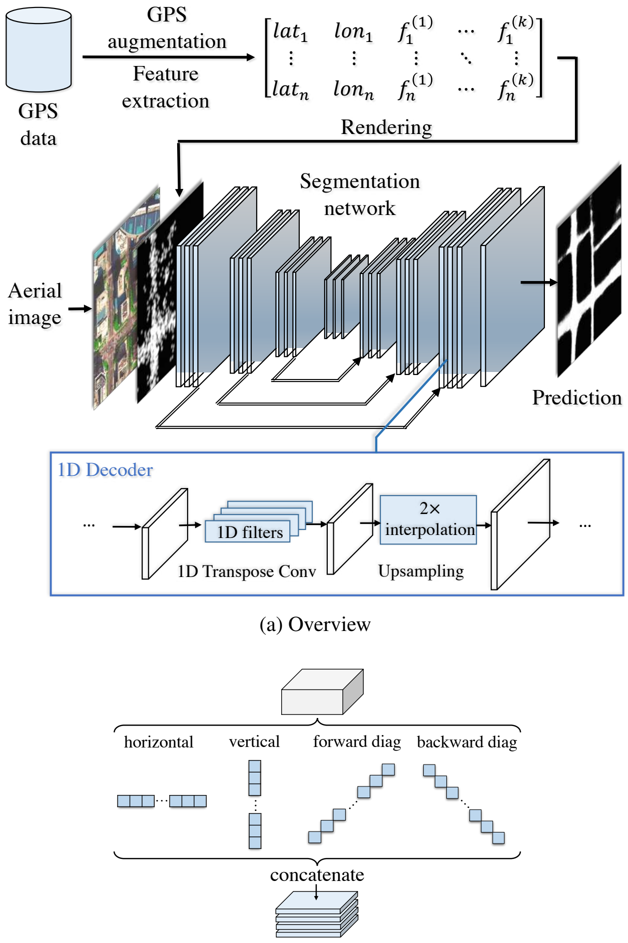
Resolution	Dataset	
	Beijing	Shanghai
lat/lon (degree)	1/100,000	1/60,000 or 1/10,000
speed (km/h)	1 or 2	1 or 2
bearing (degree)	3 or 10	2 or 45

Different vehicles may use different GPS receivers with different settings. Figure 3 shows the log scale distributions of sampling intervals and device-measured speed of our datasets. It is obvious from the figure that different taxis have different sampling interval settings, most notably at 10, 60, 180, and 300 seconds for the Beijing dataset, and 16 and 61 seconds for the Shanghai dataset. The speed distribution shows two layers of outline curves because the samples have different speed resolutions, most commonly 1 and 2 km/h. Therefore the outer layer corresponds to even numbers and the inner layer corresponds to odd numbers. Latitude, longitude, and bearing have different resolutions too, summarized in Table 1. Most Beijing taxis are at 10^{-5} degree, or roughly 1 m. Shanghai taxis have resolutions as low as 10^{-4} degree, or roughly 10 m. Our satellite imagery has a resolution of 50 cm/pixel that is higher than our GPS data. Therefore, there is the mosaic effect where some pixels have no GPS samples and some pixels may have multiple samples as the data quantity increases; see Fig. 2 zoomed in. Crowdsourced GPS data are cheap and abundant. There can be multiple datasets for just one area. We must develop a model robust against different data characteristics so there is no need to retrain the model with new datasets.

4. Method

By rendering GPS data as new input layers like RGB channels, our method applies to all existing CNN-based semantic segmentation networks. GPS data augmentation prevents overfitting and gives a robust model against different GPS data characteristics. Replacing the 3×3 transpose convolution in the decoder by 1D transpose convolution gives better accuracy, called 1D decoder for the rest of this paper.

4.1. Architecture



(a) Overview

(b) 1D transpose convolution block

Figure 4: Network architecture

In DeepGlobe'18 road extraction challenge [10], all top teams used variants of fully convolutional net for pixel segmentation [27], e.g., U-Net [24] and DeepLab [8]. The winner team modified LinkNet [7] that is very similar to U-net, by adding dilated convolutions to accommodate much larger input size and to produce more structured output, called D-LinkNet [39]. We propose to render GPS data as images and concatenate with RGB channels as the input to the segmentation net; see Fig. 4a. Therefore our method applies to most existing segmentation networks. More specifically, based on the input image coordinates, we query database for the relevant GPS data in the area and get for example n samples where each sample i has coordinates lat_i, lon_i and other features like sampling interval and vehicle speed $f_i^{(1)}, ...f_i^{(k)}$. Like image augmentation frequently employed in image processing training, we augment GPS data to prevent overfitting. Afterwards, we render the data as one or multiple image layers based on the number of features used.

Unlike natural objects, roads are thin, long, and often straight. The square kernels that dominate most CNN architectures have square receptive fields that are more suitable for natural objects of bulk shapes. For roads, it takes a very large square to cover a long straight road, where many pixels can be irrelevant. The 1D filters are more aligned with road shapes. We find that these 1D filters are most effective in the decoder block as replacements for 3×3 transpose convolutions, as the lower portion of Fig. 4a depicts.

Let $\boldsymbol{k} \in \mathbb{R}^{2r+1}$ denotes the 1D transpose convolution filter of size $2r+1$, and $\boldsymbol{y_I} \in \mathbb{R}^{H\times W}$ be the result of 1D transpose convolution of input $\boldsymbol{x} \in \mathbb{R}^{H\times W}$ and the filter $\boldsymbol{k}$ at direction $\boldsymbol{I} = (I_h, I_w)$. We have

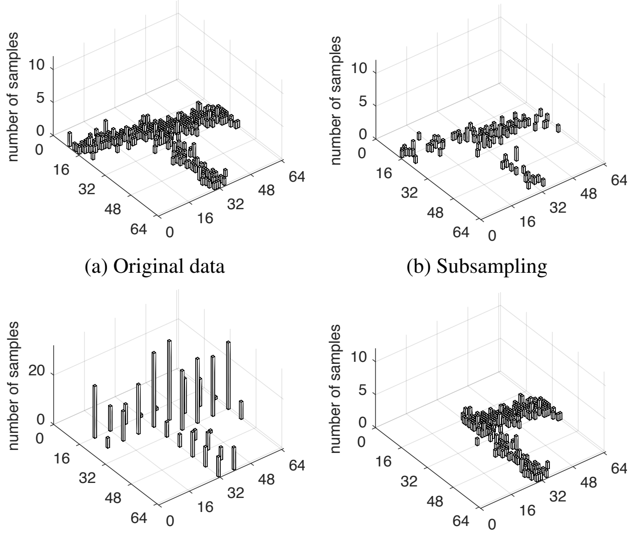
$$\begin{aligned} \boldsymbol{y_I}[i,j] = (\boldsymbol{x} *^T \boldsymbol{k})_{\boldsymbol{I}} = \\ \sum_{t=-r}^{r} \boldsymbol{x}[i + I_h t, j + I_w t] \cdot \boldsymbol{k}[r-t] \end{aligned} \tag{1}$$

where $\boldsymbol{x} *^T \boldsymbol{k}$ is the transpose convolution operation, and $\boldsymbol{I}$ is the direction indicator vector of the 1D filter, which takes four values $(0,1), (1,0), (1,1), (-1,1)$ for horizontal, vertical, forward diagonal, and backward diagonal transpose convolution, respectively, shown in Fig. 4b.

We set $r = 4$ and thus each 1D filter has 9 parameters, the same as the 3×3 transpose convolution filter. Our 1D decoder replaces each of the 3×3 transpose convolution layer by four sets of 1D filters of the four directions in concatenation. The number of 1D filters in each set is 1/4 of the total number of 3×3 filters. Therefore, the total number of network parameters and the computation cost remain the same. Our 1D decoder is especially effective against roads with sparse GPS samples, e.g., residential roads, by reducing gaps in the prediction.

4.2. Data Augmentation

Deep CNNs are very complex models prone to overfitting, especially for GPS data that is relatively simple and well structured. In our experiments, adding a GPS layer without any data augmentation leads to a superficial model which enhances RGB-based predictions wherever GPS data is dense, and suppresses the prediction wherever there is no



(a) Original data

(b) Subsampling

(c) Sub-resolution

(d) Random omission

Figure 5: GPS data augmentation

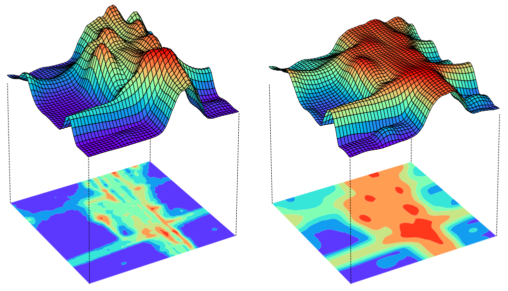
GPS data. In addition, the model is very sensitive to GPS quantity and quality. For example, if we remove the GPS input altogether, the prediction is a lot worse than the model trained with RGB image input only. We develop the following augmentation methods to prevent overfitting.

- Randomly subsample the input GPS data
- Reduce the resolution of the input GPS data by a random factor, called sub-resolution hereafter
- Random perturbation of the GPS data
- Omitting a random area of GPS data

Figure 5 illustrates some of these augmentation techniques. Figure 5a shows the GPS samples on a 64 x 64 image patch. The height of the bars indicates the number of samples projected to the same pixel, between zero to three in this case. Figure 5b takes a random 60% of samples from Fig. 5a. Figure 5c reduces all samples to 1/8 of their original resolution such that the samples are aggregated to a small set of pixels. Many GPS data have low resolution either because of inferior GPS receivers used or because of privacy protection. In addition, sub-resolution leads to much higher values for the remaining pixels than the original data, which is similar to the case of larger GPS quantities. The model trained with sub-resolution handles unseen larger amount of GPS data better in our experiments. Figure 5d omits samples on the left 32 x 32 square.

4.3. Rendering

After augmentation, we must render the GPS data as an image layer to concatenate with the RGB image input.



(a) Linear scale

(b) Log scale

Figure 6: Gaussian kernel rendering of Fig. 2a

There are many different ways to render the image. For example in Fig. 2, we render a pixel white if and only if there is at least one GPS sample projected to it. This method works with small datasets only. As the GPS quantity increases, noise spreads and too many pixels will be white, like Fig. 2a.

Instead of a binary image, we can use a greyscale image where the number at each pixel indicates the number of samples projected to it, therefore road pixels will have higher values than noise pixels as the quantity increases. Inspired by Kernel Density Estimation (KDE) frequently used in road inferencing from GPS data [9], we can also render the GPS data with Gaussian kernel smoothing. Figure 6a is the Gaussian kernel rendering of Fig. 2a. Because of data disparity between highways and residential roads, log scale could make infrequently traveled roads more prominent. For example in Fig. 6b, the horizontal road at the bottom becomes much more visible than in the linear scale.

When there is a limited quantity of GPS data but the sampling frequency is high, adding a line segment between consecutive samples helps [15], which is another way to render GPS data. In our case, these line segments often shortcut intersections and curves because of low sampling frequency, and therefore do not improve results in our experiments. Our 1D decoder has similar effect at roads with sparse samples, and they are not affected by sampling intervals.

Other GPS measurements can be useful for road extraction. We render these measurements as separate input layers. More specifically, the pixel values of the interval, speed, and bearing layers are the average sampling interval, average speed, and average sinusoid of the bearing for all the samples projected to the pixel, respectively.

5. Experiments

We experiment with the satellite imagery and our GPS datasets from two cities, and report our results here.

Datasets For satellite imagery, we crawled 350 images in Beijing and 50 images in Shanghai from Gaode map [2]. All these images are 1024 x 1024 in size and 50 cm/pixel in resolution, a total area of about 100 km^2. Like the Deep-

Table 2: Different input and model combinations

input	method	IoU (%) on *test* set	
		plain	1D decoder
GPS	KDE [9]	34.06	-
	DeepLab (v3+) [8]	47.65	-
	U-Net [24]	43.63	48.10
	Res U-Net [38]	45.33	48.52
	LinkNet [7]	49.98	**51.06**
	D-LinkNet [39]	48.46	49.95
image	DeepLab (v3+)	43.40	-
	U-Net	51.85	52.10
	Res U-Net	50.26	51.77
	LinkNet	53.96	54.84
	D-LinkNet	54.42	**55.15**
image + GPS	DeepLab (v3+)	50.81	-
	U-Net	53.22	54.88
	Res U-Net	52.29	54.24
	LinkNet	57.48	57.89
	D-LinkNet	56.96	**57.96**

Globe dataset, we manually created the training labels by masking out road pixels in the images. We choose the same input image size as the DeepGlobe data set for the convenience of comparison. It is also an appropriate size because a smaller one would lose the context and a larger size may not fit in GPU memory. The DeepGlobe dataset is for much larger areas but we do not have GPS data in the areas for experiments. Some other research work used large datasets by rendering OSM road vectors with fixed width, typically for developed countries [5, 18]. Roads in developing countries vary in width more significantly, and misalignments are prevalent on OSM. Therefore we have to label road pixels manually. Nevertheless, our dataset is among the largest in research work that do not use DeepGlobe datasets or OSM labels [16, 40].

Our GPS datasets are taxi and bus samples that include timestamp, latitude, longitude, speed, bearing, and vehicle status flags. As discussed in Section 3, our GPS datasets are from different devices with varying sampling rates and different resolutions for the measurements.

Similar to the competition and the other research work, we use the intersection over union (IoU) as the main evaluation criteria, and report the average IoU among all test image patches. We randomly split our dataset into three partitions, 70% for training, 10% for validation, and the rest 20% for testing. Other than the last experiment that evaluates the ability for our model to predict new areas, we use only the Beijing satellite images and GPS dataset for training and testing.

Models Our GPS rendering method applies to all existing segmentation models. Here we choose DeepLab, two variants of U-Net, and two variants of LinkNet to evaluate. The two variants of U-Net are the original one and the one with ResNet style encoder and decoder, denoted as Res U-Net. The two variants of LinkNet are the original one and D-LinkNet that achieved top performance in the DeepGlobe challenge. For road extraction using GPS input only, we also add KDE method for comparison since it was among the best using traditional machine learning techniques [15].

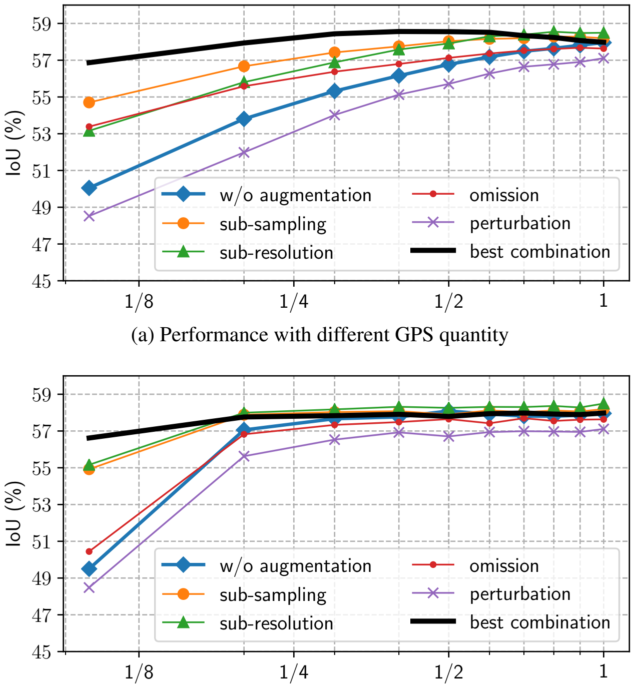
Baseline Our first experiment takes the GPS input alone; see the top section of Table 2. For the KDE method, since we measure IoU only and do not extract road centerlines, we simply pick the best kernel size and the threshold to binarize the Gaussian smoothed image. The results show that deep neural nets perform much better than the KDE method to extract roads from GPS data only. Our 1D decoder is very useful against relatively shallow neural nets, and give about 1 % increase against more complex models. LinkNet shows the best result here. Although D-LinkNet performed better in the DeepGlobe challenge, its additional complexity over LinkNet leads to more severe overfitting of the relatively simple GPS data. We do not apply 1D decoder to DeepLab since it uses a bi-linear interpolation decoder without any transpose convolution.

Next we examine the performance of the different segmentation models with the satellite image input only; see the second section of Table 2. The result is consistent with the numbers reported in the DeepGlobe challenge, where D-LinkNet is slightly better than the other models [39]. The best IoU in our test is lower than the number in the challenge because the Beijing area is more challenging than the rural areas and towns used in the challenge. Many roads are completely blocked by tree canopy in the old city center area, and the road boundaries are not easy to define with the prevalent express/local/bike way systems. DeepLab has the worst performance among the models we use. Visually examining the output reveals much coarser borders than in the other model output, likely due to the bi-linear interpolation decoder instead of transpose convolution used.

Finally, with both the image and the GPS input, D-LinkNet remains the top performer. Here the largest performance gain for the additional GPS input is DeepLab. For the other models that already perform relatively well on the image input, the performance gain is about 2 %, and the 1D decoder adds about another 1 %.

Augmentation Figure 7 shows the effectiveness of our GPS data augmentation. Figure 7a and Figure 7b are the performance of different augmentation techniques with a subset of input data and a reduced resolution of input data, respectively. With our data augmentation, our model not only performs much better with degraded GPS data input, but also gains about 0.5% over the top-of-the-line performance.

Rendering As described in Section 4.3, Fig. 8 shows the performance of Gaussian kernel rendering with different kernel sizes and different rendering scale. We also



(a) Performance with different GPS quantity

(b) Performance with different GPS resolution

Figure 7: GPS data augmentation results (D-LinkNet with image+GPS input)

experimented with various combination of GPS measurements, sampling interval, vehicle speed, and vehicle bearing. Adding another input layer of sampling interval alone gives the best performance gain. Based on these results, we use two input layers for the GPS data for the rest of the experiments, Gaussian kernel rendering of the GPS samples with kernel size three and the sampling interval channel.

Table 3 is the overall performance gain with various improvements over the baseline using the image input only. Altogether we achieved 4.76% performance gain.

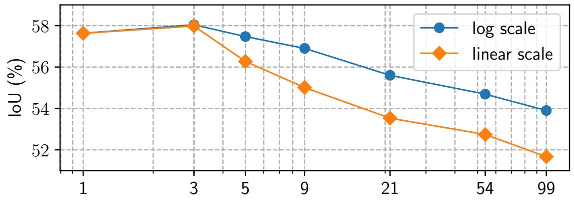
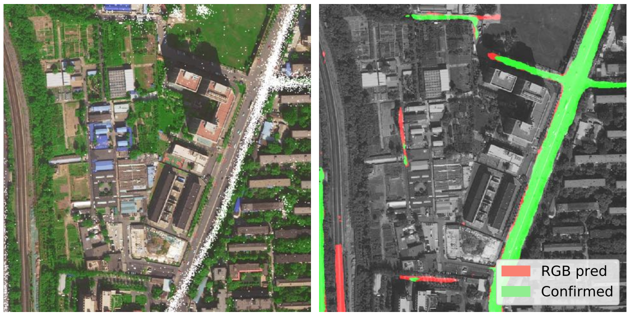


Figure 8: Rendering with different Gaussian kernel sizes

GPS as verification As discussed in Section 1, local verification is often required for mapping. Figure 9 shows how the crowdsourced GPS data can serve the verification purpose without local survey. Here the green pixels are high confidence predictions by both the image-only input and the image + GPS input, while red pixels are high confidence predictions by image-only input but low confidence predictions when GPS input is added. Map matching could give additional confidence by matching GPS traces to roads by topology [21], which is beyond the scope of this paper.

Table 3: Using GPS features and data augmentation

settings (all using D-LinkNet)	IoU (%)
image	54.42
image + GPS	56.96
image + GPS + 1D decoder	57.96
image + GPS + 1D decoder + augment.	58.55
image + GPS + interval + 1D decoder	58.55
image + GPS + interval + 1D decoder + augment.	**59.18**



(a) GPS samples over satellite image

(b) Roads confirmed by GPS

Figure 9: Road verification using GPS data

New testing area Table 4 is the testing results with our Shanghai dataset using different training data and methods. Despite the different GPS data characteristics, it is evident that prediction with additional GPS input is more resilient in the new domain, 18.9% IoU drop for the model trained with both datasets instead of 31.6% for the model trained with image input only. The performance gain is enhanced when employing the GPS data augmentation, confirming its effect against overfitting.

Table 4: Shanghai testing dataset results

train	method	IoU(%)	relative
Beijing + Shanghai	GPS	44.88	–
	image	55.76	–
	image + GPS (w/o augment)	59.30	–
	image + GPS (w/ augment)	**60.00**	–
Beijing	GPS	42.82	-4.6%
	image	38.16	-31.6%
	image + GPS (w/o augment)	44.57	-24.9%
	image + GPS (w/ augment)	**48.69**	-18.9%

Qualitative results Figure 10 visualizes the road extraction results of different methods in different testing areas of Beijing and Shanghai, trained using Beijing dataset only. Overall, prediction using GPS data only largely matches the sample distribution. With the image input only,

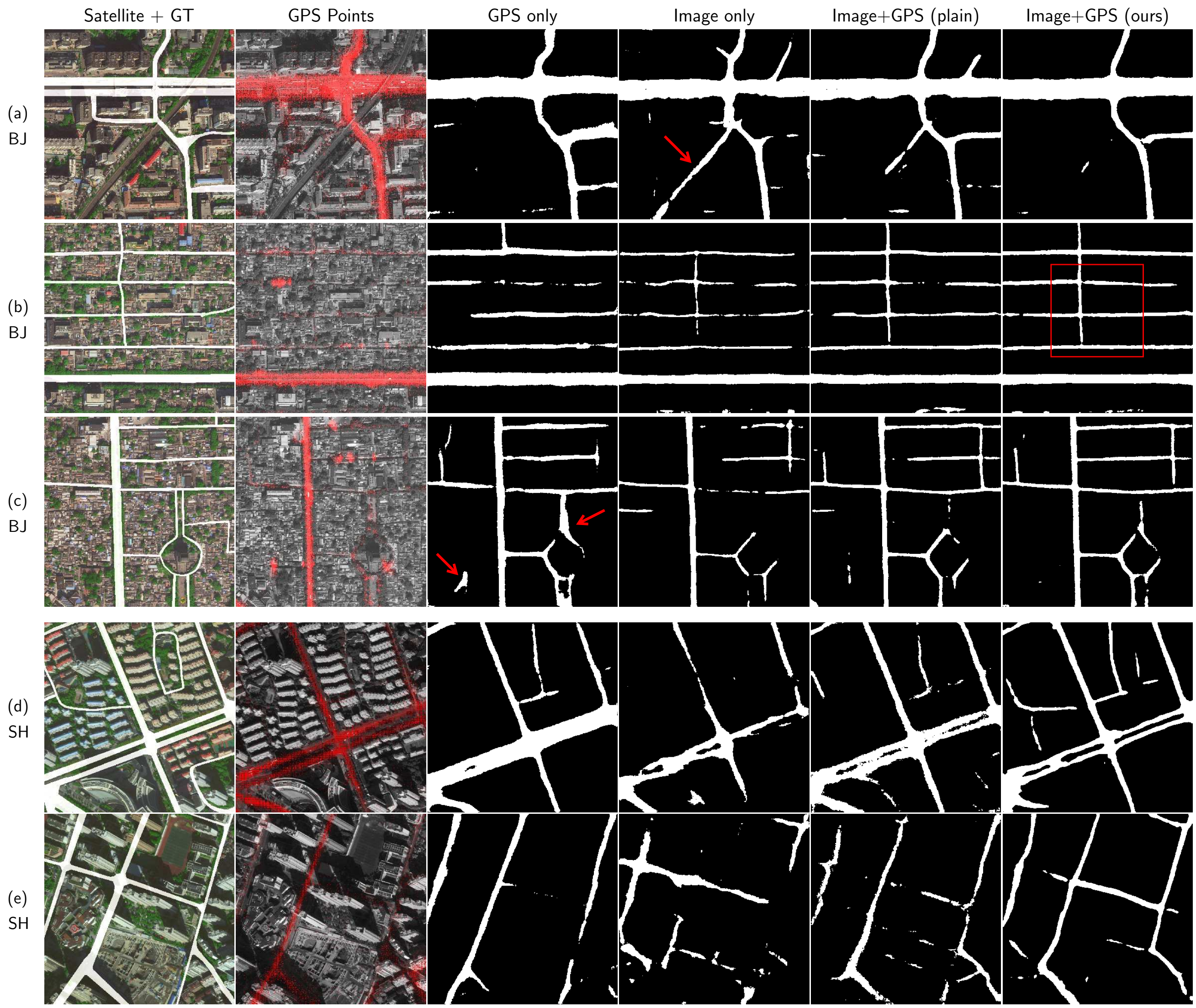


Figure 10: Prediction results using different methods on Beijing and Shanghai testing datasets trained on Beijing dataset only

occlusion and other image issues can cause poor performance. Both image and GPS input give the best results, and our enhancement techniques give a bit cleaner output. As examples, the areas pointed by red arrows show false positives removed with our model using GPS data. The one in the first row is a railway and the one in the third row is from GPS noise. The red square shows an area with dense tree canopy and relatively sparse GPS samples. Only the combination of image and GPS data extracts a relatively complete road network.

6. Conclusion

With large-scale crowdsourced GPS datasets, fusing GPS data with aerial image input gives much better road segmentation results than using images or GPS data alone with deep neural net models. Inspired by image augmentation techniques, our GPS data augmentation is very effective against overfitting, and thus our method performs much better in new testings areas than other models. In our experiences, aerial imagery works best for residential roads detection because they are relatively simple, numerous, and infrequently traveled. In contrast, GPS data can recover arterial roads with ease even for complicated highway systems and under severe image occlusion. Therefore, the two data sources well complement each other for road extraction tasks.

Acknowledgement

We thank anonymous reviewers for valuable feedback. This research is supported by NSFC General Program 41771481, Shanghai Science and Technology Commission program 17511104502, and a gift fund from Facebook.

References

[1] AI-assisted road tracing. wiki.openstreetmap.org/wiki/AI-Assisted_Road_Tracing. 1, 2

[2] Gaode Map. www.amap.com. 5

[3] OpenStreetMap. www.openstreetmap.org. 1, 2

[4] S. Aksoy, B. Ozdemir, S. Eckert, F. Kayitakire, M. Pesarasi, O. Aytekin, C. C. Borel, J. Cech, E. Christophe, S. Duzgun, et al. Performance evaluation of building detection and digital surface model extraction algorithms: Outcomes of the prrs 2008 algorithm performance contest. In *IAPR Workshop on Pattern Recognition in Remote Sensing*. IEEE, 2008. 1, 2

[5] F. Bastani, S. He, S. Abbar, M. Alizadeh, H. Balakrishnan, S. Chawla, S. Madden, and D. DeWitt. Roadtracer: Automatic extraction of road networks from aerial images. In *Computer Vision and Pattern Recognition (CVPR)*, 2018. 2, 6

[6] J. Biagioni and J. Eriksson. Map inference in the face of noise and disparity. In *SIGSPATIAL Conference on Geographic Information Systems (GIS)*, 2012. 2

[7] A. Chaurasia and E. Culurciello. Linknet: Exploiting encoder representations for efficient semantic segmentation. In *Visual Communications and Image Processing (VCIP)*, 2017. 4, 6

[8] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2018. 2, 4, 6

[9] J. J. Davies, A. R. Beresford, and A. Hopper. Scalable, distributed, real-time map generation. *IEEE Pervasive Computing*, 5(4):47–54, 2006. 2, 5, 6

[10] I. Demir, K. Koperski, D. Lindenbaum, G. Pang, J. Huang, S. Basu, F. Hughes, D. Tuia, and R. Raskar. Deepglobe 2018: A challenge to parse the earth through satellite images. In *Computer Vision and Pattern Recognition (CVPR) Workshops*, 2018. 2, 4

[11] B. Huang, K. Lu, N. Audebert, A. Khalel, Y. Tarabalka, J. Malof, A. Boulch, B. Le Saux, L. Collins, K. Bradbury, et al. Large-scale semantic classification: outcome of the first year of inria aerial image labeling benchmark. In *IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, 2018. 2

[12] S. Karagiorgou, D. Pfoser, and D. Skoutas. A layered approach for more robust generation of road network maps from vehicle tracking data. *ACM Transactions on Spatial Algorithms and Systems (TSAS)*, 3(1):3, 2017. 2

[13] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 2

[14] P. Li, Y. Zang, C. Wang, J. Li, M. Cheng, L. Luo, and Y. Yu. Road network extraction via deep learning and line integral convolution. In *IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, 2016. 2

[15] X. Liu, J. Biagioni, J. Eriksson, Y. Wang, G. Forman, and Y. Zhu. Mining large-scale, sparse gps traces for map inference: comparison of approaches. In *ACM SIGKDD international conference on Knowledge discovery and data mining*, 2012. 2, 5, 6

[16] G. Máttyus, W. Luo, and R. Urtasun. Deeproadmapper: Extracting road topology from aerial images. In *The IEEE International Conference on Computer Vision (ICCV)*, 2017. 2, 6

[17] H. Mayer, S. Hinz, U. Bacher, and E. Baltsavias. A test of automatic road extraction approaches. *International Archives of Photogrammetry, Remote Sensing, and Spatial Information Sciences*, 36(3):209–214, 2006. 1, 2

[18] V. Mnih. *Machine Learning for Aerial Image Labeling*. PhD thesis, University of Toronto, 2013. 2, 6

[19] V. Mnih and G. E. Hinton. Learning to label aerial images from noisy data. In *International conference on machine learning (ICML)*, 2012. 1, 2

[20] A. J. Mosinska, P. Marquez Neila, M. Kozinski, and P. Fua. Beyond the pixel-wise loss for topology-aware delineation. In *Computer Vision and Pattern Recognition (CVPR)*, 2018. 2

[21] P. Newson and J. Krumm. Hidden markov map matching through noise and sparseness. In *SIGSPATIAL Conference on Geographic Information Systems (GIS)*, 2009. 7

[22] D. Patel. Osm at facebook. State of the Map, 2018. 1, 2

[23] S. Rogers, P. Langley, and C. Wilson. Mining gps data to augment road models. In *ACM SIGKDD international conference on Knowledge discovery and data mining*, 1999. 2

[24] O. Ronneberger, P. Fischer, and T. 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. 2, 4, 6

[25] S. Schrödl, S. Schrödl, K. Wagstaff, S. Rogers, P. Langley, and C. Wilson. Mining GPS traces for map refinement. *Data Mining Knowledge Discovery*, 9(1):59–87, 2004. 2

[26] Z. Shan, H. Wu, W. Sun, and B. Zheng. Cobweb: a robust map update system using gps trajectories. In *ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp)*, 2015. 2

[27] E. Shelhamer, J. Long, and T. Darrell. Fully convolutional networks for semantic segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39(4):640–651, 2017. 2, 4

[28] T. Sun, Z. Di, and Y. Wang. Combining satellite imagery and gps data for road extraction. 2018. SIGSPATIAL Conference on Geographic Information Systems (GIS) workshop. 2

[29] M. Trifunovic. Robot tracers - extraction and classification at scale using & cntk. State of the Map, 2018. 1, 2

[30] F. van Diggelen. Gnns accuracy: Lies, damn lies, and statistics. *GPS World*, pages 26–32, 2007. 3

[31] M. Volpi and D. Tuia. Dense semantic labeling of subdecimeter resolution images with convolutional neural networks. *IEEE Transactions on Geoscience and Remote Sensing*, 55(2):881–893, 2017. 2

[32] S. Wang, M. Bai, G. Mattyus, H. Chu, W. Luo, B. Yang, J. Liang, J. Cheverie, S. Fidler, and R. Urtasun. Torontocity: Seeing the world with a million eyes. In *International Conference on Computer Vision (ICCV)*, 2017. 2

[33] Y. Wang. Scaling Maps at Facebook. In *SIGSPATIAL Conference on Geographic Information Systems (GIS)*, 2016. keynote. 2

[34] Y. Wang, X. Liu, H. Wei, G. Forman, C. Chen, and Y. Zhu. Crowdatlas: Self-updating maps for cloud and personal use. In *International Conference on Mobile Systems, Applications, and Services (MobiSys)*, 2013. 1, 2

[35] G.-S. Xia, X. Bai, J. Ding, Z. Zhu, S. Belongie, J. Luo, M. Datcu, M. Pelillo, and L. Zhang. Dota: A large-scale dataset for object detection in aerial images. In *Computer Vision and Pattern Recognition (CVPR)*, 2018. 2

[36] N. Yokoya, P. Ghamisi, J. Xia, S. Sukhanov, R. Heremans, I. Tankoyeu, B. Bechtel, B. Le Saux, G. Moser, and D. Tuia. Open data for global multimodal land use classification: Outcome of the 2017 ieee grss data fusion contest. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 11(5):1363–1377, 2018. 2

[37] J. Yuan and A. M. Cheriyadat. Image feature based gps trace filtering for road network generation and road segmentation. *Machine Vision and Applications*, 27(1):1–12, 2016. 2

[38] Z. Zhengxin, L. Qingjie, and W. Yunhong. Road extraction by deep residual u-net. In *IEEE GEOSCIENCE AND REMOTE SENSING LETTERS*, 2017. 6

[39] L. Zhou, C. Zhang, and M. Wu. D-linknet: Linknet with pretrained encoder and dilated convolution for high resolution satellite imagery road extraction. In *Computer Vision and Pattern Recognition (CVPR) Workshops*, 2018. 4, 6

[40] X. X. Zhu, D. Tuia, L. Mou, G.-S. Xia, L. Zhang, F. Xu, and F. Fraundorfer. Deep learning in remote sensing: a comprehensive review and list of resources. *IEEE Geoscience and Remote Sensing Magazine*, 5(4):8–36, 2017. 2, 6