ABSTRACT
Road network and building footprint extraction is essential for many applications such as updating maps, traffic regulations, city planning, ride-hailing, disaster response etc. Mapping road networks is currently both expensive and labor-intensive. Recently, improvements in image segmentation through the application of deep neural networks has shown promising results in extracting road segments from large scale, high resolution satellite imagery. However, significant challenges remain due to lack of enough labeled training data needed to build models for industry grade applications. In this paper, we propose a two-stage transfer learning technique to improve robustness of semantic segmentation for satellite images that leverages noisy pseudo ground truth masks obtained automatically (without human labor) from crowd-sourced OpenStreetMap (OSM) data. We further propose Pyramid Pooling-LinkNet (PP-LinkNet), an improved deep neural network for segmentation that uses focal loss, poly learning rate, and context module. We demonstrate the strengths of our approach through evaluations done on three popular datasets over two tasks, namely, road extraction and building footprint detection. Specifically, we obtain 78.19% meanIoU on SpaceNet building footprint dataset, 67.03% and 77.11% on the road topology metric on SpaceNet and DeepGlobe road extraction dataset, respectively.

KEYWORDS
Mapping application; Remote Sensing; Hyperspectral Imaging; Transfer Learning; Road network; Building footprint; Multi-stage training; PP-LinkNet

1 INTRODUCTION
Maps are common heritage for humanity. Creating and updating maps is an important task with many potential applications in disaster recovery, geospatial search, urban planning, ride-hailing industry, etc [9]. Today, map features such as roads, building footprints, and points of interest are primarily created through manual techniques. We address the problem of extracting road networks and building footprints automatically from satellite imagery in this paper.

Maps of roads and buildings are expensive to build and maintain. Although modern cartography involves using satellite and aerial imagery along with GPS traces, using these data to update maps involves having humans analyze the data in a time consuming process, and thus maps of rapidly growing cities (where infrastructure is constantly under construction) are still often inaccurate and incomplete outside the urban core. Current approaches to map inference via satellite images [10, 39] involve extensive annotations of images that is both tedious and time consuming.

Currently, we have two large public datasets containing satellite images of road networks with annotations: i) SpaceNet [39] and DeepGlobe [10]. SpaceNet challenge 3 road extraction [39] only provides 2779 training satellite images of Las Vegas, Paris, Shanghai,
2 RELATED WORK

Datasets for road network. Two large datasets for road network extraction from satellite imagery (i.e., SpaceNet [39] and DeepGlobe [10]). They differ mainly in the format of annotation data – while SpaceNet dataset provides centerlines for roads, DeepGlobe provides pixel labeling of roads. Several published works use proprietary datasets that are not available in the public domain for benchmarking, e.g., [4, 25, 27, 41].

Semantic segmentation models for road network extraction. Inspired by successes of deep learning in computer vision [15, 16, 24, 33, 37], there have been few attempts in addressing the problem of road extraction from satellite images using deep learning based semantic segmentation models. D-LinkNet [43], the winner model of DeepGlobe road extraction challenge, utilizes LinkNet[7], UNet[34], ResNet [16] and dilated convolution[8] components model and improvises using a post-processing step to refine the segmentation output to a road network graph. DeepRoadMapper [25] developed a model based on ResNet [16] and FCN [35], and a technique to infer missing connections from segmentation results. On the other hand, RoadTracer [4] uses convolutional neural network (CNN) to predict the walking angle of the sliding window on the input image to decide on the input image for the next iteration.

Our work is similar to [2], where the authors use OSM data along with satellite imagery to predict semantic map features (e.g., surfaces, buildings, vegetables, trees, cars, etc). On the contrary, our work utilizes OSM as pseudo ground truth data only in the first stage of our two-stage transfer learning.

3 GENERATING NOISY PSEUDO GROUND TRUTH

In this section, we describe our approach to collect a large-scale satellite image dataset along with its pseudo ground truth to train road network and building footprint segmentation model in the first stage.

3.1 Rasterization of OSM data into pseudo (noisy) ground truth image

In order to avoid laborious annotation work and to create a scalable approach towards creating training data we relied on OpenStreetMap (OSM)[19], a large community driven map. OSM data contains many useful metadata information such as type of roads, number of lanes, width, surface, bridge, tunnel, etc. Our source of satellite imagery is from different sources of remotely sensed data. The procedure to extract ground truth from OSM data is as follows: i) collect satellite images for the area of interest, i) render the road network from OSM using TileMill software. This rendering process is called the rasterization process in the geographical information system (GIS) community. There are few GIS softwares that can do rasterization such as TileMill, QGIS, ArcGIS, etc. Furthermore, we also develop our rasterization process using OpenCV. Specifically, our process is to draw lines with certain widths corresponding to locations of roads in the image. Figure 1 shows that our rasterization process would produce similar pseudo ground truth image with the result from TileMill software. We use our rasterization process to convert OpenStreetMap data into pseudo ground truth mask image. Top row: rasterization using TileMill software. Bottom row: our rasterization using OpenCV from the corresponding road network graph. Best viewed in color.
We build our training data in the first stage by combining data from SpaceNet [39], DeepGlobe [10], and our internal data. Both SpaceNet and DeepGlobe datasets have high resolution satellite images (i.e., 0.3-0.5m). We remove some imperfect images of SpaceNet [39], which has zero signal due to sensor’s errors. The DeepGlobe dataset contains cloud-free, high resolution satellite images. Furthermore, we also collect our own data from additional sources such as internal data, EarthExplorer [38], etc. Currently, our dataset has two resolutions 1024 × 1024 and 1300 × 1300. We decide to build the dataset with high resolution satellite images because we consider the narrowness, connectivity, complexity, and long span of roads. Eventually, we utilize 48091 images and their corresponding noisy pseudo ground truth masks to train the model of road network segmentation in the first phase.

4 METHOD

We formulate our task of extracting roads from satellite images as a binary semantic image segmentation problem, where for each input satellite image, we predict if a pixel belongs to class 1 (Road) or class 0 (non-road). Distinct from datasets for object detection such as PASCAL VOC 2012 [31] and MS-COCO [22], satellite images need to be processed at a high resolution (e.g., 1024 × 1024). This compels the need to design our network with efficient memory optimisations.

4.1 Network Architecture Design

Figure 3 shows an overview of our proposed network architecture Pyramid Pooling-LinkNet (PP-LinkNet). Our network design is inspired by the encoder-decoder architecture such as Unet [34]. We received input images at resolutions of 1024 × 1024, hence, we emphasize on efficient network structure. We choose LinkNet [7] as our base network, similar to D-LinkNet [43]. Our encoder is ResNet34 [16] pre-trained on ImageNet dataset. RestNet34 is originally trained to classify images of mid-resolution 256 × 256, whereas, our task is to segment roads and building footprints from higher resolution imagery 1024 × 1024. Therefore, the network needs to adapt encoder layers to new domain of input. Our decoder includes bottleneck blocks, and layers of decoder up-sample feature size to have symmetric sizes with encoder’s layers. The bottleneck block of a decoder consists of a transposed convolutional layer between two convolutional layers with (1 × 1)-kernels (see Figure 3). Our decoder is initialized with random parameters. In the middle of our network is global contextual prior, pyramid pooling (PP) module, which has been empirically proven to be effective in PSPNet [42]. Pyramid pooling module has no parameters, and proven to have better performance than the context module in D-LinkNet in our experiments.

4.2 Improvements of semantic segmentation system for road network extraction

Two-stage transfer learning. In the first phase, we train our PP-LinkNet on noisy pseudo ground truth generated from OSM (Sect. 3) to let the network learn basic visual features and knowledge of the new domain (i.e., aerial or satellite imagery). For the second phase, we propose a simple transfer learning procedure to fine-tune the trained model one more time with high-quality well-annotated data.

Class imbalance. It is obvious that the number of non-road pixels is much bigger than the number of road pixels. This extreme

\[
(lon, lat) = T \cdot (r + 0.5, c + 0.5, 1),
\]

where \((lon, lat)\) is the longitude, latitude according to the image coordinate \((c, r)\). From equation 1, given the longitude, latitude of a node in OSM road networks, we can infer its image coordinates from the inverse matrix of the transformation matrix \(T\).

There are three main advantages of our approach of using noisy pseudo ground truth images as part of training data. First, annotating pixel-to-pixel for large image size of satellite images (e.g., 1024 × 1024) is an exhaustive task. Generating ground truth through rasterizing OSM data is efficient so that we can scale the number of training images up easily. Second, the pseudo ground truth of satellite imagery is generally available for most locations on earth. We only need to know the geographical address of the locations to rasterize the pseudo ground truth masks. Finally, the accuracy of OSM ground truth is enough to train a moderate model in the first phase. Hence, we devised a second phase in order to fine-tune our model with high quality annotated ground truth.

Figure 2 shows an example of a pair of input and its rasterized pseudo ground truth mask in Las Vegas city from SpaceNet dataset. Although the structure of road network in the example is correct with the satellite image, its pixel-to-pixel matching is limited. Sometimes, the pseudo ground truth is smaller or bigger than the actual road, hence, it is called pseudo ground truth.

3.2 Image collection

We build our training data in the first stage by combining data from SpaceNet [39], DeepGlobe [10], and our internal data. Both SpaceNet and DeepGlobe datasets have high resolution satellite images (i.e., 0.3-0.5m). We remove some imperfect images of SpaceNet [39], which has zero signal due to sensor’s errors. The DeepGlobe dataset contains cloud-free, high resolution satellite images. Furthermore, we also collect our own data from additional sources such as internal data, EarthExplorer [38], etc. Currently, our dataset has two resolutions 1024 × 1024 and 1300 × 1300. We decide to build the dataset with high resolution satellite images because we consider the narrowness, connectivity, complexity, and long span of roads. Eventually, we utilize 48091 images and their corresponding noisy pseudo ground truth masks to train the model of road network segmentation in the first phase.

4 METHOD

We formulate our task of extracting roads from satellite images as a binary semantic image segmentation problem, where for each input satellite image, we predict if a pixel belongs to class 1 (Road) or class 0 (non-road). Distinct from datasets for object detection such as PASCAL VOC 2012 [31] and MS-COCO [22], satellite images need to be processed at a high resolution (e.g., 1024 × 1024). This compels the need to design our network with efficient memory optimisations.

4.1 Network Architecture Design

Figure 3 shows an overview of our proposed network architecture Pyramid Pooling-LinkNet (PP-LinkNet). Our network design is inspired by the encoder-decoder architecture such as Unet [34]. We received input images at resolutions of 1024 × 1024, hence, we emphasize on efficient network structure. We choose LinkNet [7] as our base network, similar to D-LinkNet [43]. Our encoder is ResNet34 [16] pre-trained on ImageNet dataset. RestNet34 is originally trained to classify images of mid-resolution 256 × 256, whereas, our task is to segment roads and building footprints from higher resolution imagery 1024 × 1024. Therefore, the network needs to adapt encoder layers to new domain of input. Our decoder includes bottleneck blocks, and layers of decoder up-sample feature size to have symmetric sizes with encoder’s layers. The bottleneck block of a decoder consists of a transposed convolutional layer between two convolutional layers with (1 × 1)-kernels (see Figure 3). Our decoder is initialized with random parameters. In the middle of our network is global contextual prior, pyramid pooling (PP) module, which has been empirically proven to be effective in PSPNet [42]. Pyramid pooling module has no parameters, and proven to have better performance than the context module in D-LinkNet in our experiments.

4.2 Improvements of semantic segmentation system for road network extraction

Two-stage transfer learning. In the first phase, we train our PP-LinkNet on noisy pseudo ground truth generated from OSM (Sect. 3) to let the network learn basic visual features and knowledge of the new domain (i.e., aerial or satellite imagery). For the second phase, we propose a simple transfer learning procedure to fine-tune the trained model one more time with high-quality well-annotated data.

Class imbalance. It is obvious that the number of non-road pixels is much bigger than the number of road pixels. This extreme
class imbalance causes our network to pay more attention to background pixels to learn weights of semantic segmentation systems. Focal loss has been proposed in [21] to tackle class imbalanced foreground and background sampled bounding boxes in object detection pipelines. Focal loss for each pixel $FL(p_{i,t})$ and for an image $FL$ can be expressed as

$$FL(p_{i,t}) = -\alpha_t (1 - p_{i,t})^\gamma \log(p_{i,t})$$

$$FL = \sum_t FL(p_{i,t}),$$

where $p_{i,t}$ is the model’s estimated probability for the class with label $y = t$. Specifically, we have two class, non-road and road, $(0, 1)$ and $p \in [0, 1]$ is the estimated probability for the class with label $y = 1$, then

$$p_t = \begin{cases} p, & \text{if } y = 1 \\ 1 - p, & \text{otherwise.} \end{cases}$$

From different experiments, we found that $-\alpha_t, \gamma = 0.5$ is the best parameters for the focal loss function $FL(p_{i,t})$.

Proposed in [28], Dice coefficient loss, denoted as $DL$, can be written as

$$DL = 1 - \frac{2 \sum_{i=1}^{N} p_t g_t + \epsilon} {\sum_{i=1}^{N} p_t^2 + \sum_{i=1}^{N} g_t^2 + \epsilon},$$

where $N$ is the number of pixels in image, $p_t \in [0, 1]$ is the predicted probability of a pixel, and $g_t \in [0, 1]$ is the binary ground truth of the pixel. The $\epsilon$ term is a small constant to ensure the loss function stability by preventing numerical issues of dividing by 0. The Dice loss originated from Sørensen–Dice coefficient [1], a statistic used for comparing similarity between two samples. Optimally, when we minimize $DL$ loss, an optimizer would increase $p_t$ for foreground pixels ($i.e., g_t = 1$) and decrease $p_t$ for background pixels ($i.e., g_t = 1$). Dice loss does not suffer from class imbalance because contributions of individual pixels only influence numerator or denominator term, not the whole loss function.

**Poly learning rate policy.** Similar to [8, 23], we also explore a “poly” learning rate policy ($i.e.,$ the learning rate is equal to the base learning rate multiplied by $(1 - \frac{\text{iter}}{\text{max}_\text{iter}})^\text{power}$) with $\text{power} = 0.9$ for training DL-Net.

**Context module.** One improvement of PP-LinkNet compared to D-LinkNet is the pyramid pooling context module. We integrate the pyramid pooling module [42] into LinkNet [7] between encoder and decoder part. Through our empirical experiments, we found that the pyramid pooling module with no parameters is effective in improving performance of our model, hence, we named it PP-LinkNet ($i.e.,$ PyramidPooling-LinkNet).

## 5 EXPERIMENTS

### 5.1 Dataset

**DeepGlobe road extraction** [10]. Road extraction task in DeepGlobe challenge has 6226 training, 1243 validation, and 1101 testing images, with image resolution of $1024 \times 1024$. The ground resolution is 50 cm/pixel. However, we don’t have annotations for validation and testing images. We follow [36] to split the training image set into 4696 images for training and 1530 for validation.

**SpaceNet challenge 2 building footprints** [10]. The dataset includes 24,586 images of size 650×650 containing 302,701 building footprints across Las Vegas, Paris, Shanghai, Khartoum. The ground resolution is 30 cm/pixel. We split the training set of dataset into 80%/20% for training/test, equivalently to, 8472 images for training and 2121 for testing. We conduct experiments on only 3 RGB channels of multi-spectral images, and on full resolution 650 × 650 of input images. More information about F1 evaluation metric of the dataset can be found in DeepGlobe building detection task [10].

**SpaceNet challenge 3 road extraction** [10]. The dataset includes 2780 images, and, following [36], we split the dataset into 2213 images for training and 567 images for testing. The ground truth is represented by line strings, indicating center-lines of the roads. The image has a resolution of $1300 \times 1300$, and its ground resolution is 30cm/pixel. Unlike [5], we train our model on larger image crop $1024 \times 1024$, not $512 \times 512$. As human errors, we always...
have some mistakes in annotations of three datasets, but we leave it for future analyses.

5.2 Goal and Metrics

Pixel based metric. The semantic segmentation algorithm is expected to predict a mask for a test image (i.e., road and non-road labels). Similar to [10], we deploy the standard VOC intersection-over-union IoU metric [12] defined as Equ. 5,

$$IoU_i = \frac{TP_i}{TP_i + FP_i + FN_i},$$

where $TP_i$ is the number of pixels that are correctly predicted as road pixel, $FP_i$ is the number of pixels that are wrongly predicted as road pixel, and $FN_i$ is the number of pixels that are wrongly predicted as non-road pixel for image $i$. The final evaluation score is computed as the mean IoU of all $n$ images (i.e., Equ. 6),

$$mIoU = \frac{1}{n} \sum_{i=1}^{n} IoU_i.$$  \hfill (6)

Graph based metric. To measure the difference between estimated road network and ground truth graph, we use Average Path Length Similarity (APLS) in [39], as evaluation metric. The APLS metric sums up the differences in optimal path lengths between nodes in the estimated graph $G'$ and the ground truth graph $G$. The APLPS metric scales from 0 (poor) to 1 (perfect),

$$C = 1 - \frac{1}{N} \sum_{(a,b)} \min \left\{ 1, \frac{|L(a,b) - L(a',b')|}{L(a,b)} \right\},$$  \hfill (7)

where $N$, number of unique paths, while, $L(a,b)$, length of path $(a,b)$. The node $a'$ denotes the projected node (i.e., into the ground truth graph) in the proposal graph closest to the node $a$ of the ground truth graph. We also have a similar metric from the ground truth graph $G$ into the proposal graph $G'$. If path $(a',b')$ does not exist, then the maximum penalty would be used. Hence, a missing or redundant segment would also be penalized.

5.3 Implementation Details

Baseline models. We compare our method with recent state-of-the-art methods that are all based on deep learning and leverage pre-trained models on large-scale image classification dataset such as ImageNet [11]. There are several algorithms proposed for semantic segmentation such as FCN [35], U-Net [10], DeepLab [8], SegNet [3], PSPNet [42], LinkNet [7], ENet [32], etc. However, there are few works adapted for road network extraction from satellite images. In this work, we present our baseline results based on three networks TernausNetV2 [20], PSPNet [42], and D-LinkNet [43]. TernausNetV2 [20] and D-LinkNet [43] are the winner models for building detection task, and road extraction task respectively in the DeepGlobe challenge 2018 [10]. Both tasks share similar characteristics which are dense prediction of per-pixel labels (i.e., building polygons and road surface). An improved version of PSPNet [42] is the winning submission of the 2017 Large Scale Scene Understanding (LSUN) challenge on scene semantic segmentation.

Training procedures. There are interesting evidences that a bigger crop size of input image would boost performance of semantic segmentation problem [6, 40]. Our PP-LinkNet is designed for dealing with high resolution satellite images of $1024 \times 1024$ pixels. In order to have fair comparisons, we also train three baseline algorithms with the same resolution of input images. For the three baseline algorithms, we follow training as detailed by the authors. For example, for D-LinkNet [43], we use Adam optimizer with learning rate $2e^{-4}$, and reduced by 5 for 3 times when training loss decreases slowly. We train our PP-LinkNet in 90 epochs with learning rate $2e^{-4}$ and batch size of 9 in our experiments. TernausNetV2 consumes large GPU memory for a big crop size of $1024 \times 1024$, hence, we only train TernausNetV2 networks with half-precision floating-point numbers 16-bits $^\dagger$ [30] (i.e., fp16) and a batch size of 2.

5.4 Results

Ablation study. Table 1 shows the ablation study for our PP-LinkNet designs and the two-stage transfer learning. As can be observed, both DLinkNet34 and PSPNet outperforms TernausNetV2 fp16. It might be due to the fact that the mixed precision training is not yet standardized. DLinkNet34 has better performance than PSPNet (67.42% vs. 63.38% mIOU), although DLinkNet34 and PSPNet have the same ResNet34 backbone network.

Figure 4a shows the training behavior of three loss functions. Interestingly, as can be seen, the focal loss obtains the smallest training loss compared to the BCE and Dice loss. The Dice loss is harder than the BCE and focal loss to be optimized in DLinkNet framework, but it also shows a monotonically decreasing loss values. Table 1 presents meanIoU and APLS scores on test set of Deepglobe road extraction dataset [10]. It is consistent with the training loss behaviours that our result for the focal loss model (i.e., 67.82%) is higher the BCE (i.e., 67.21%), and Dice loss model (i.e., 66.31%) respectively. Figure 5 illustrates the difference in confidence map of the BCE+Dice loss model compared to the focal loss model. With the same input image, the BCE+Dice model pays attention to main roads, while the focal loss model is able to put confidence on both main and small roads. This observation is also reflected in

\footnote{https://github.com/NVIDIA/apex}
APLS scores with a big margin. Our final model with focal loss has improved on the original DLinkNet34 [43] with BCE+Dice loss 5.26% on APLS scores (i.e., 77.11% vs. 71.85%). We also argue that APLS score is more important meanIoU score, because it is inherent to connectivity of road network graphs. From following experiments, we use the focal loss for training PP-LinkNet because it is shown as a single effective loss function for the road extraction task.

Figure 4b shows the training behavior of the poly learning rate policy. The poly learning rate policy is more effective to optimize the deep semantic segmentation system than the reduce-on-plateau learning rate policy. As a result, we improve DLinkNet34 with poly learning rate 1.67% meanIoU from 67.82% to 69.49%. We will use the poly learning rate for the following experiments.

Additionally, Table 1 shows the result of PP-LinkNet34 model with pyramid pooling (PP) context module is better than D-LinkNet34 model with dilation context module (i.e., 69.87% vs. 69.49% in meanIoU, 76.04% vs. 75.71% in APLS). Compared to the original D-LinkNet34, we obtain a larger margin of performance (i.e., 69.87% vs. 67.42% in meanIoU, 76.04% vs. 71.85%) and it suggests that other proposed modifications also have compound effects with the pyramid pooling context module. In other experiments in SpaceNet road network and building footprint dataset (see Table 2, 3), our PP context module also has consistently better performance than dilation context module. Figure 6 shows that our context module can help PP-LinkNet to infer road segments that are difficult to predict when PP-LinkNet does not have the context module.

Finally, our two-stage training further boost performance of the network especially in terms of APLS scores. PP-LinkNet two-stage improves on the first-stage one 3.34% in APLS scores (i.e., 77.11% vs. 73.77%), and also achieve the best results over all model variations.

**Road network extraction.** Table 2 shows our results on SpaceNet and DeepGlobe road extraction dataset. With our two-stage training, we are able to improve quantitative metrics over PP-LinkNet one-stage training over two datasets. We achieve larger gaps of improvements in APLS scores in both dataset. APLS score involves topological measures, hence, it means that our two-stage approach boosts the ability of PP-LinkNet to detect road connectivity.

Furthermore, Table 2 also shows that our PP-LinkNet obtains the new state-of-the-art performance on both SpaceNet and DeepGlobe dataset. Notice that SpaceNet road dataset does not have

---

**Table 1: Ablation study of different design choices for our PP-LinkNet architecture with two-stage training on DeepGlobe road extraction dataset.** First block: baseline results; second block: various modifications on D-LinkNet34; third block: evolution of PP-LinkNet; forth block: comparisons between two-stage vs. one-stage training.

| Method | PP-LinkNet34 first-stage (on OSM data) | PP-LinkNet34 two-stage (Ours) |
|--------|---------------------------------------|------------------------------|
| mIoU   | 68.12                                 | 70.53                        |
| APLS (%) | 73.77                                 | 77.11                        |

| Method | DeepRoadMapper (segmentation) | DeepRoadMapper (full) | RoadCNN (segmentation) | MatAN | Topology Loss (with BCE) | Topology Loss (with SoftIoU) | LinkNet34 | D-LinkNet34 (Ours reproduced) | Improved road connectivity | Self-supervision Coach Mask | Sat2Graph |
|--------|-------------------------------|-----------------------|------------------------|-------|--------------------------|-------------------------------|------------|-----------------------------|---------------------------|--------------------------|-----------|
| mIoU   | 59.99                         | 54.25                 | 59.99                  | 67.99 | 58.34                    | 67.99                         | 59.99      | 55.90                       | 63.75                     | 77.0         | 64.43     |
| APLS (%) | 62.58                         | 65.56                 | 62.58                  | 69.65 | 64.95                    | 69.65                         | 62.75      | 67.53                       | 67.21                     | 76.8         | 76.8      |

**Table 2: Our results of our PP-LinkNet models on the road network extraction task, and comparison of our technique with the state-of-the-art road network extraction techniques on SpaceNet and DeepGlobe road extraction dataset.**

| Method | SpaceNet mIoU | APLS (%) | DeepGlobe mIoU | APLS (%) |
|--------|---------------|----------|----------------|----------|
| PP-LinkNet34 (Ours) | 55.50 | 65.94 | 69.87 | 76.04 |
| PP-LinkNet34 two-stage (Ours) | 56.43 | 67.03 | 78.53 | 77.11 |

**Table 3: Results of our PP-LinkNet model on building footprint detection task.** Top half: different variations of our PP-LinkNet; Bottom half: our reproduced results from other works.

| Method | meanIoU (%) | F1 score (%) |
|--------|-------------|--------------|
| PP-LinkNet34 two-stage (Ours) | 78.19 | 68.92 |
| PP-LinkNet34 first-stage (OSM data) | 44.00 | 25.33 |
| PP-LinkNet34 (Ours) | 76.62 | 67.04 |
| D-LinkNet34 (Ours reproduced) | 76.21 | 66.79 |
we don’t use multi-spectral channels and ensembles of multiple deep learning models. Figure 7 demonstrates some visualizations of the winner of SpaceNet building detection. Furthermore, our PP-LinkNet model has performance on par with upper-bound performance on the dataset. We don’t have training cates that pyramid pooling module captures global scene context better than DLinkNet34’s context module. We don’t have training script for TernausNetV2 trained on both our train and test set, therefore, its released model’s result would approximate the upper-bound performance on the dataset. Additionally, our PP-LinkNet model has performance on par with the winner of SpaceNet building detection, although we don’t use multi-spectral channels and ensembles of multiple deep learning models. Figure 7 demonstrates some visualizations of our segmentation result mask. It shows that our model can pick out building footprint reasonably well (i.e., third column in Figure 7).

6 DISCUSSIONS

Domain adaptation is an important procedure for adapting the models of supervised learning into domains different with the trained environment. Often, the new domain (e.g., aerial, medical imagery) has small data or small annotated data. Our two-stage transfer learning is beneficial in adapting supervised learning trained with big data into new domains with small data.

Our two-stage transfer learning is also closely related to training deep learning models in the revival era of deep learning. At that time, we pre-train each layer of Restricted Boltzmann Machine in unsupervised fashion, and then, we fine-tune the whole network for different tasks. In our work, with the prevalence of supervised learning techniques, computing resources, and data, we first pre-train the whole deep model on pseudo-groundtruth (noisy) data, then fine-tune it on high-quality annotated data.

7 CONCLUSION

This paper introduces a new procedure to generate and utilize per-pixel pseudo ground truth labels for extracting road network and building footprint in satellite images. The procedure would allow us to fine-tune semantic segmentation systems on less amount of annotated satellite imagery. Furthermore, we show the new state-of-the-art evaluation results of our PP-LinkNet system on three challenging remote sensing datasets.
