remote sensing

MDPI

Article

Learn to Extract Building Outline from Misaligned Annotation through Nearest Feature Selector

Yuxuan Wang [1,†], Guangming Wu [1,*,†], Yimin Guo [1], Yifei Huang [2] and Ryosuke Shibasaki [1]

[1] Center for Spatial Information Science, The University of Tokyo, Kashiwa 277-8568, Japan; yuxuan@csis.u-tokyo.ac.jp (Y.W.); guo.ym@csis.u-tokyo.ac.jp (Y.G.); shiba@csis.u-tokyo.ac.jp (R.S.)
[2] Institute of Industrial Science, The University of Tokyo, Tokyo 153-8505, Japan; hyf@iis.u-tokyo.ac.jp
* Correspondence: huster-wgm@csis.u-tokyo.ac.jp; Tel.: +81-04-7136-4390
† These authors contributed equally to this work.

Received: 4 July 2020; Accepted: 20 August 2020; Published: 23 August 2020

Abstract: For efficient building outline extraction, many algorithms, including unsupervised or supervised, have been proposed over the past decades. In recent years, due to the rapid development of the convolutional neural networks, especially fully convolutional networks, building extraction is treated as a semantic segmentation task that deals with the extremely biased positive pixels. The state-of-the-art methods, either through direct or indirect approaches, are mainly focused on better network design. The shifts and rotations, which are coarsely presented in manually created annotations, have long been ignored. Due to the limited number of positive samples, the misalignment will significantly reduce the correctness of pixel-to-pixel loss that might lead to a gradient explosion. To overcome this, we propose a nearest feature selector (NFS) to dynamically re-align the prediction and slightly misaligned annotations. The NFS can be seamlessly appended to existing loss functions and prevent misleading by the errors or misalignment of annotations. Experiments on a large scale aerial image dataset with centered buildings and corresponding building outlines indicate that the additional NFS brings higher performance when compared to existing naive loss functions. In the classic L1 loss, the addition of NFS gains increments of 8.8% of f1-score, 8.9% of kappa coefficient, and 9.8% of Jaccard index, respectively.

Keywords: deep convolutional networks; outline extraction; misalignments; nearest feature selector

1. Introduction

The rooftops of buildings are dominant features in urban satellite or aerial imagery. For many remote sensing applications, such as slum mapping [1], urban planning [2], and solar panel capacity analysis [3], the spatial distributions and temporal renews of buildings are critical. These information are collected from labor-intensive and time-consuming field surveys [4]. For analyses in the city or country scale, especially in developing countries, a robust and cost-efficient method for automatic building extraction is preferred.

Over the past decades, many algorithms have been proposed [5]. These methods are verified by datasets of various types (e.g., imagery or point cloud), scales (e.g., city or country), resolutions (e.g., centimeter or meter), or spectrums (e.g., visible light, or multispectral) [6–10]. Based on whether sampled ground truths are required, existing building outline extraction methods can be classified into two categories: (i) unsupervised and (ii) supervised methods.

1.1. Unsupervised Methods

For most unsupervised methods, building outlines are extracted using thresholding pixel values or histograms [11], edge detectors [12], and region techniques [13,14]. Because of their simplicity,

these methods do not require additional training data and are fast. However, when applied to residential areas with complex backgrounds, some artifacts and noises are inevitable in the extracted building outlines.

1.2. Supervised Methods

Unlike unsupervised methods, supervised methods extract building outlines from the images through patterns learned from ground truths. By learning from correct examples, supervised methods typically performed better in terms of both generalization and precision [15–17].

In the early stages, a two-stage approach that combines handcrafted descriptors for feature extraction [18–21] and classifiers for categorizing [22–24] are adopted in supervised methods. Because of the separation, an optimal combination of both the feature descriptor and classifier is difficult to achieve. Rather than the two-stage approach, convolutional neural network (CNN) methods enable a unified feature extraction and classification through sequential convolutional and fully connected layers [25,26]. Initially, CNN-based methods are constructed in a patch-by-patch manner that predicts the class of a pixel through the surrounding patch [27]. Subsequently, fully convolutional networks (FCNs) are introduced to reduce memory costs and improve computational efficiency through sequential convolutional, subsampling, and upsampling operations [28,29]. Because of information loss caused by subsampling and upsampling operations, the prediction results of classic FCN models often present blurred edges. Hence, advanced FCN-based methods using various strategies have been proposed, such as unpooling [30], deconvolution [31], skip connections [32,33], multi-constraints [34], and stacking [35]. Among FCN-based methods, two different approaches exist: (a) indirect and (b) direct approaches.

1.2.1. Indirect Approach

In the indirect approach, instead of extracting the building outline directly from the input aerial or satellite image, semantic maps are first generated. The outlines on top of those maps are computed consequently. Because the outlines are derived from segmentation output, the final accuracy relies significantly on the robustness of semantic segmentation.

In principle, all FCN-based methods mentioned above can be used for indirect building outline extraction. However, owing to the sensitivity of the outline/boundary, training with only semantic information typically results in an inconsistent outline or boundary. To prevent this, BR-Net [36] utilizes a modified U-Net, and a multitask framework to generate predictions for semantic maps and building outlines based on a consistent feature representation from a shared backend.

1.2.2. Direct Approach

Unlike the indirect approach, the direct approach extracts the building outlines directly from the input aerial or satellite images. Compared with the indirect approach, the direct approach learns the extraction pattern directly from the ground truth outline that preserves a higher fidelity. In the direct approach, building outline extraction is considered a segmentation or pixel-level classification problem that involves extremely biased data [37]. In recent years, some advanced FCN-based models, such as RSRCNN [38], ResUNet [39], and D-LinkNet [40] have been proposed for better outline extractions.

However, these models focus on deeper network architectures to better utilize the feature representation capability of hidden layers. Furthermore, regardless of how these models generate predictions, their loss functions are computed directly from the pixel-to-pixel similarity of the ground truth. Owing to the extremely biased distribution of positive and negative pixels, the gradient explosion during training becomes a severe problem. Additionally, because of occasional human errors, several or tens of pixel misalignments will inevitably occur between the annotation and the corresponding aerial image. Owing to the much fewer positive pixels of the building outline, the pixel-to-pixel losses are extremely sensitive to these misalignments.

Hence, we propose a nearest feature selector (NFS) module, enabling a dynamic re-alignment between the ground truth and prediction. A dynamic matching between the ground truth and

prediction is performed at every iteration to determine the matched position. Subsequently, the overlapped areas of both the ground truth and prediction are used for further loss computation. Because the NFS is used for the upper stream, it can be seamlessly integrated into all existing loss functions. The effectiveness of the proposed NFS module is demonstrated using a VHR image dataset [36] located in New Zealand (see Section 2.1). In comparative experiments, under different loss functions, the addition of the NFS indicates significantly higher values of the f1-score, Jaccard index [41], and kappa coefficient [42].

The main contributions of this study can are as follows:

- We design a fully convolutional network framework for direct building outline extraction from aerial imagery.
- We propose the nearest feature selector(NFS) module to dynamically re-align the prediction and annotation to avoid misleading by slightly misaligned annotations.
- We analyze the effectiveness of the NFS with different loss functions to understand its effects on the performances of deep CNN models.

The rest of the paper is organized as follows: At first, we introduce the materials and methods used for this research in the Section 2. Then, we present the learning curves and quantitative and qualitative results in the Section 3. Subsequently, we illustrate our discussion and conclusion in the Sections 4 and 5, respectively.

2. Material and Method

2.1. Data

To evaluate the performance of different methods, a research area located in Christchurch, New Zealand, is selected. The original aerial imagery, as well as annotated building polygons, are hosted by the Land Information of New Zealand (LINZ) (https://data.linz.govt.nz/layer/53413-nz-building-outlines-pilot/). The aerial images are in a spatial resolution of 0.075. Prior to performing our experiments, we evenly partition the study area into two areas for training (i.e., Figure 1a, left) and testing (i.e., Figure 1a, right), respectively. The original annotations provided by the LINZ are registered to the corresponding building grounds instead of rooftops (confirmed by visual interpretation uisng QGIS GUI (https://qgis.org/)). For accurate outline extraction, we manually adjust vectorized building outlines to ensure that all building polygons and aerial rooftops are roughly registered (i.e., Figure 1b). Because of the huge amount of buildings and occasional human errors, sub-pixel or several pixel misalignments will be inevitable. Thus, we have to train the models with imperfect "ground truth".

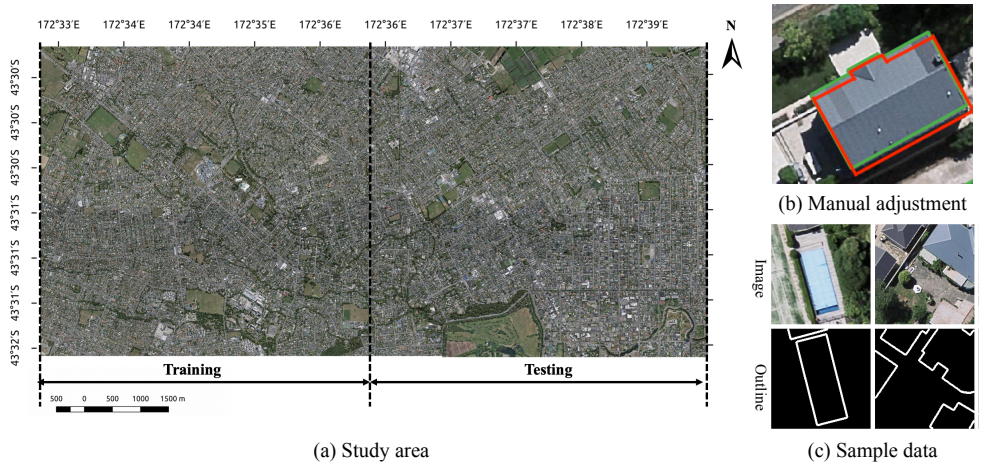


Figure 1. (**a**) Aerial imagery of the study area ranging from 172°33′E to 172°40′E and 43°30′S to 43°32′S, encompassing approximately 32 km^2. (**b**) Manual adjustment of provided annotation (e.g., from Red to Green polygon). (**c**) Sample pairs of the extracted patches.

As shown in Figure 1a, the study area is covered mainly by residential buildings with sparsely distributed factories, trees, and lakes. From training and testing areas, 16,635 and 14,834 patches are extracted. The size of the patch is 224 × 224 pixels. As shown in Figure 1c, within each pair of the patches, there are buildings in the center area.

2.2. Methodology

In this study, we are expected to correctly train and evaluate a model using imperfect annotation. Due to the inevitable misalignments, values of the loss functions or metrics, which are directly computed by the pixel-to-pixel comparison of the prediction and annotation, are inaccurate. To avoid this, we introduce the nearest feature selector (NFS) module to perform similarity selection during training and testing stages.

As shown in Figure 2, at the training phase, the NFS is applied to prediction and imperfect annotation to generate aligned prediction and annotation for accurate loss estimation and proper back-propagation. As for the testing phase, the NFS is applied to prediction and imperfect annotation to generate aligned prediction and annotation that can be used for reliable accuracy analysis. Since the NFS is applied to select the most paired overlap, it can avoid misalignments in the ground truth and produce a more reliable accuracy or prediction error.

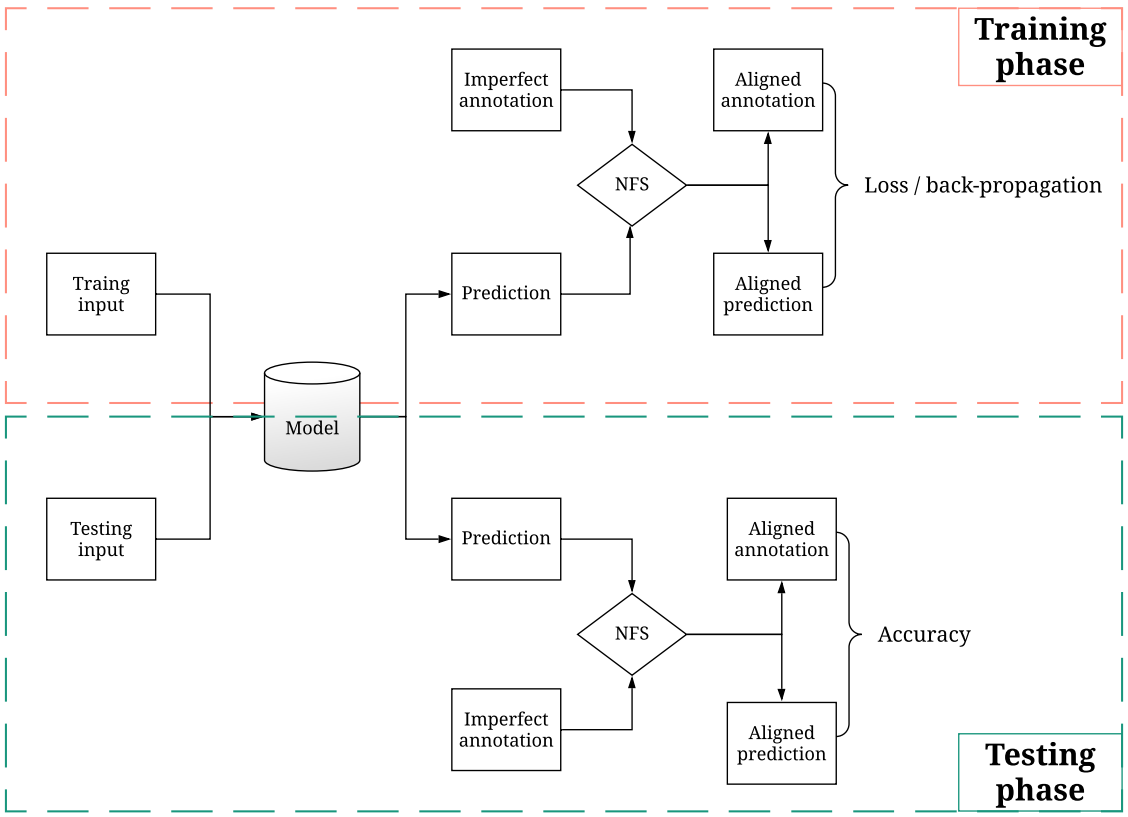


Figure 2. Experimental design for model training and evaluation under imperfect annotation. The proposed nearest feature selector(NFS) is applied to perform similarity selection during training and testing stages.

Figure 3 presents the workflow for building outline extraction. The aerial images and their corresponding building outlines are partitioned into two sets for training and testing. Through several cycles of training and validation, the hyperparameters, including batch size, the number of iterations, random seed, and initial learning rate were determined and optimized using the basic model (i.e., SegNet + L1 loss). Subsequently, the predictions generated by the optimized models are evaluated using the patches within the test set. For performance evaluations, we select three typically used

balanced metrics, i.e., the f1-score, Jaccard index, and kappa coefficient. These metrics are computed before the post-processing operations [43,44].

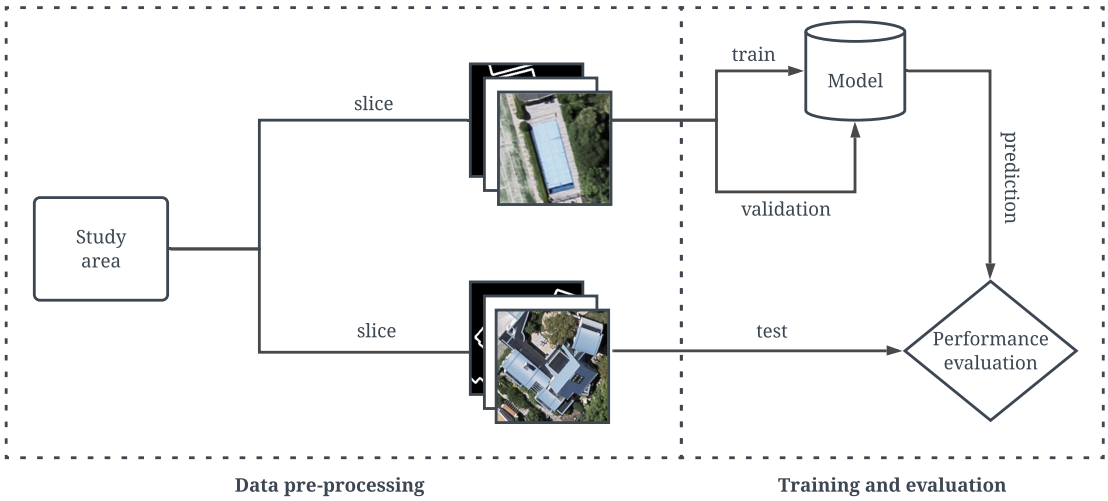


Figure 3. Experimental workflow for buidling outline extraction. Existing loss functions and proposed nearest feature selector are trained and evaluated using 224 × 224 image patches extracted from original dataset.

2.2.1. Data Preprocessing

According to the location and extent of every building polygon, a square window is applied to the centroid of the polygon to extract the corresponding image patch. Later, all patches are resized as 224 × 224 pixels. After data preprocessing, there are 16,635 and 14,834 image patches extracted from training and testing area, respectively. Since we have carefully checked the annotations, there are no negative patches to be discarded. Then, the image patches within the training area are shuffled and partitioned into two groups: training (70%), and validation (30%). Subsequently, the number of patches used for training, validation, and testing are 11,644, 4990, and 14,834, respectively.

2.2.2. Proposed Model

For an efficient building outline extraction, we utilize a modified SegNet [30] for feature extraction and the NFS to achieve a dynamic alignment between the ground truth and prediction (see Figure 4).

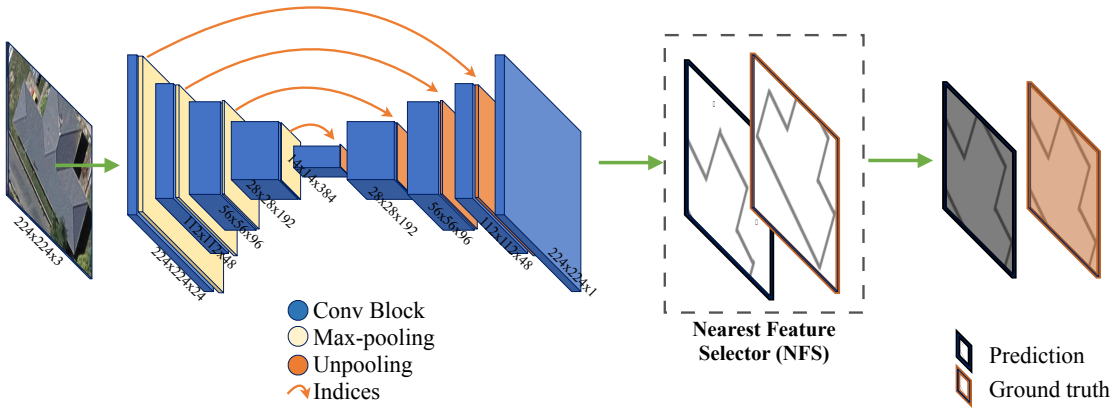


Figure 4. Overview of the proposed model. The model consists of a modified SegNet for feature extraction and the nearest feature selector (NFS) module for dynamic alignment.

- **Feature extraction**

In this study, we utilize a modified SegNet for effective feature extraction from very-high-resolution aerial images. As shown in Figure 4, the modified SegNet comprises sequential operation layers, including convolution, nonlinear activation, batch normalization, subsampling, and unpooling operations.

The convolution operation is an element-wise multiplication within a two-dimensional kernel (e.g., 3×3, or 5×5). The size of the kernel determines the receptive field and computational efficiency of the convolution operation. Owing to the complexity of the task, we set the number of kernels of the corresponding convolutional layers to [24, 48, 96, 192, 384, 192, 96, 48, 24] [34]. Subsequently, the convolution output is managed using a rectified linear unit [45], which treats all values less than zero as zeros. To accelerate network training, a batch normalization [46] layer was appended to every activation function except for the final layer. Max-pooling [47] and the corresponding unpooling [30] were used to reduce and upsample the width and height of intermediate features, respectively.

- **Nearest Feature Selector(NFS)**

Figure 5 shows the mechanisms of the NFS. The center area of the ground truth slides over the corresponding prediction along both the X- and Y-axes to generate overlaps of $\boldsymbol{X_{i,j}}$ and $\boldsymbol{Y_c}$, respectively, where i and j are the distances from the initial position. To obtain a balance between the computational efficiency and sliding field, we set the maximum values of both i and j to five. Subsequently, they were used for similarity estimation through different criteria according to the number of channels of the output.

For the prediction and ground truth containing a single channel, the classic L1 distance is used. Thus, the distance of the (i,j) overlap can be formulated as:

$$\boldsymbol{D_{i,j}} = \frac{1}{W \times H} \sum_{i=1}^{W} \sum_{j=1}^{H} ||\boldsymbol{X}_{i,j} - \boldsymbol{Y}_c|| \tag{1}$$

where $\boldsymbol{X}$ is the prediction, and $\boldsymbol{Y}$ is the corresponding ground truth. Both $\boldsymbol{X}$ and $\boldsymbol{Y}$ are $\in \boldsymbol{R}^{W \times H}$. W and H are the width and height of the corresponding output, respectively.

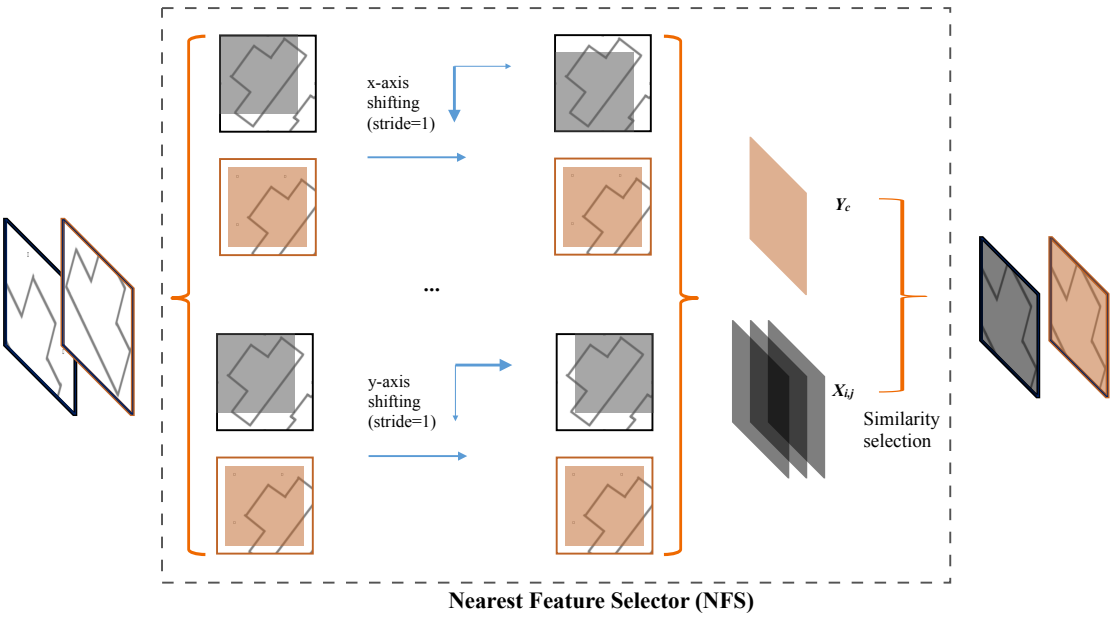


Figure 5. Overview of the nearest feature selector (NFS) module. The center area of ground truth slides over prediction along X- and Y-axes to generate overlaps that are used for similarity selection.

For the prediction and ground truth containing multiple channels, the average cosine similarity along the channels will be calculated. In such cases, the distance of overlaps can be formulated as:

$$\boldsymbol{D_{i,j}} = 1 - \frac{1}{W \times H} \sum_{i=1}^{W} \sum_{j=1}^{H} \frac{\boldsymbol{X}_{i,j} \cdot \boldsymbol{Y}_c}{||\boldsymbol{X}_{i,j}|| \times ||\boldsymbol{Y}_c||} \tag{2}$$

From all overlaps, location indices of the one with the closest distance to the ground truth is determined as:

$$(\boldsymbol{i_{min}}, \boldsymbol{j_{min}}) = \underset{i,j}{\operatorname{argmin}} \boldsymbol{D} \tag{3}$$

The nearest overlap ($\boldsymbol{X_{i_{min},j_{min}}}$) and corresponding ground truth ($\boldsymbol{Y_c}$) are selected for further final loss estimation. Four well-known loss functions, namely, L1, mean square error (MSE), binary cross-entropy (BCE) [48], and focal loss [49], are chosen in this study.

$$\boldsymbol{\mathcal{L}_{L1}} = \frac{1}{W \times H} \sum_{m=1}^{W} \sum_{n=1}^{H} ||y_{m,n} - g_{m,n}|| \tag{4}$$

$$\boldsymbol{\mathcal{L}_{MSE}} = \frac{1}{W \times H} \sum_{m=1}^{W} \sum_{n=1}^{H} (y_{m,n} - g_{m,n})^2 \tag{5}$$

where W and H represent the width and hight of the nearest overlap ($\boldsymbol{X_{i_{min},j_{min}}}$) and corresponding ground truth ($\boldsymbol{Y_c}$). The values of $y_{m,n}$ and $g_{m,n}$ are the predicted probability and ground truth, respectively.

For notational convenience, we define $p_{m,n}$:

$$p_{m,n} = \begin{cases} y_{m,n}, & \text{if } g_{m,n} = 1 \\ 1 - y_{m,n}, & \text{if } g_{m,n} = 0 \end{cases} \tag{6}$$

As compared with traditional cross-entropy, focal loss introduces a scaling factor (γ) to focus on difficult samples. Mathematically, the BCE and focal loss can be formulated as:

$$\boldsymbol{\mathcal{L}_{BCE}} = -\frac{1}{W \times H} \sum_{m=1}^{W} \sum_{n=1}^{H} log(p_{m,n}) \tag{7}$$

$$\boldsymbol{\mathcal{L}_{focal}} = -\frac{1}{W \times H} \sum_{m=1}^{W} \sum_{n=1}^{H} (1 - p_{m,n})^{\gamma} log(p_{m,n}) \tag{8}$$

Because the NFS is computed dynamically, it can be seamlessly integrated into the existing loss without further modification.

Three typically used balanced metrics, i.e., the f1-score, Jaccard index, and kappa coefficient, are used for the quantitative evaluation. Compared with unbalanced metrics such as precision and recall, the selected metrics provide a more generalized accuracy level by considering both precision and recall.

$$\boldsymbol{F1 - score} = \frac{2 \times TP}{2 \times TP + (FP + FN)} \tag{9}$$

$$\boldsymbol{Jaccard} = \frac{TP}{TP + FP + FN} \tag{10}$$

$$Pe = \frac{(TP + FN) \times (TP + FP) + (FP + TN) \times (FN + TN)}{(TP + FP + FN + FN) \times (TP + FP + FN + FN)} \tag{11}$$

$$Po = \frac{TP + TN}{TP + FP + FN + FN} \tag{12}$$

$$\boldsymbol{Kappa} = \frac{Po - Pe}{1 - Pe} \tag{13}$$

where TP, FP, FN, and TN represent the number of true positives, false positives, false negatives, and true negatives, respectively.

3. Results

Four well-known loss functions, i.e., L1, mean square error (MSE), binary cross-entropy (BCE) [48], and focal loss [49] are used in this study. The L1 and MSE can be regarded as the most classic and typically used criteria for pixel-to-pixel comparisons. The BCE is a typical loss function that increases or decreases exponentially for binary classification. The focal loss introduces a scale factor to the BCE to reduce the importance of the easy example. These loss functions were trained either with or without the NFS, separately. All experiments were performed on the same dataset and processing platform.

Three typically used balanced metrics, i.e., the f1-score, Jaccard index, and kappa coefficient, are used for the quantitative evaluation. Compared with unbalanced metrics such as precision and recall, the selected metrics provide a more generalized accuracy level by considering both precision and recall.

3.1. Learning Curves

Figure 6 shows the relative values of loss from different loss functions under the validation dataset. Among all the loss functions (i.e., L1, MSE, BCE, and focal), the loss with the NFS (i.e., +NFS) indicated a faster converging speed than those without (i.e., −NFS).

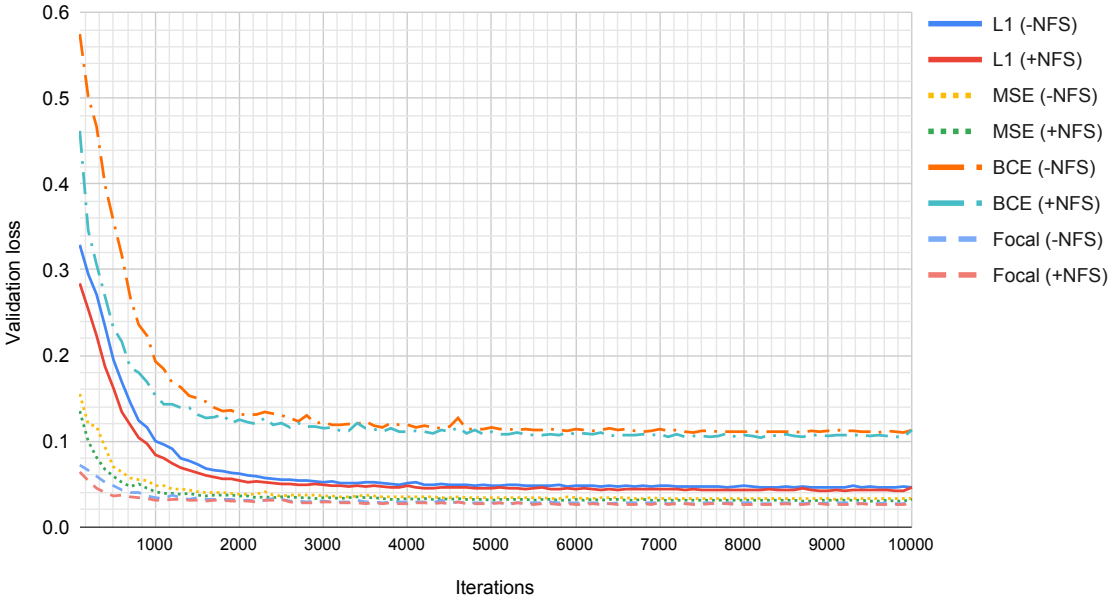


Figure 6. Trends in validation loss values over different iterations.

Figure 7 shows the trend of kappa coefficient values over various iterations from four different loss functions under the validation dataset. Among all the conditions, the focal loss trained with the proposed NFS (i.e., focal + NFS) indicates the highest kappa coefficient values in most of the iterations. By contrast, the L1 loss trained without the NFS (i.e., L1 − NFS) indicated the lowest kappa coefficient values for almost every iteration.

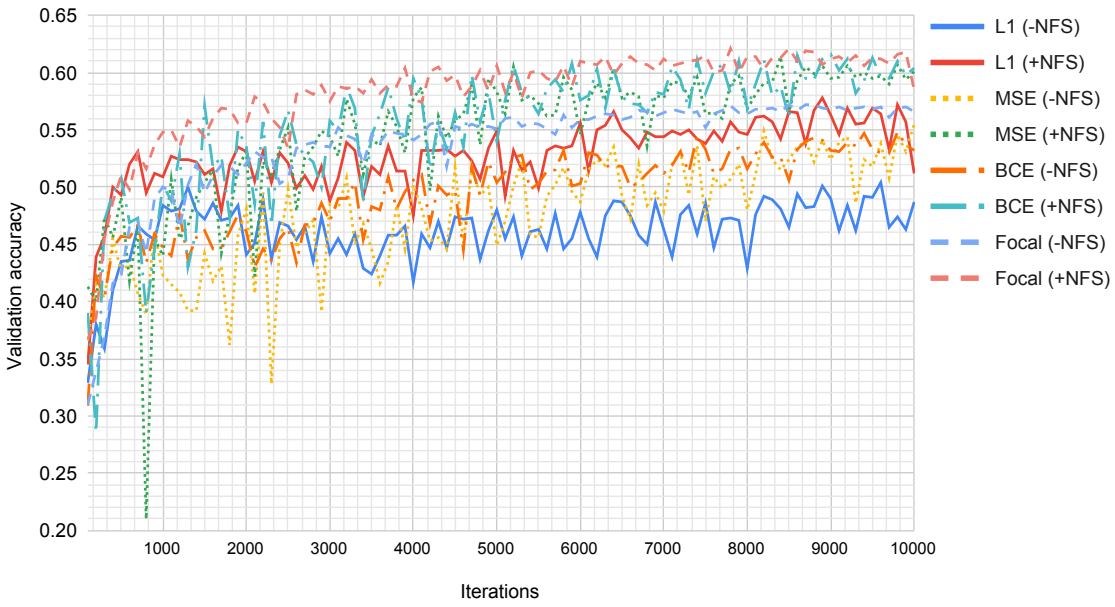


Figure 7. Trends in validation accuracy values over different iterations.

3.2. Quantitative Results

Figure 8a shows the relative performances of different loss functions under the test dataset. Among all loss functions (i.e., L1, MSE, BCE, and focal), the loss with the NFS indicates the higher values for all evaluation metrics.

Figure 8b shows the corresponding values of the evaluation metrics over various loss functions. Among four loss functions, regardless of with or without the NFS, the focal loss is generally better than BCE, MSE, and L1 loss. L1 loss without NFS (L1 − NFS) indicates the lowest values for all metrics in all conditions. The best performance is achieved by focal loss with NFS, i.e., 0.651 for f1-score, 0.490 for the Jaccard index, and 0.626 for the kappa coefficient. Under all loss functions, the addition of the NFS results in significantly higher values for all evaluation metrics. The result indicates that the proposed NFS can effectively manage the slight misalignments from the annotation and achieve better performance. Interestingly, on the weakest L1 loss, the addition of the NFS results in the most significant increments among the three evaluation metrics. The increments of the f1-score, kappa coefficient, and Jaccard index reached 8.8%, 8.9%, and 9.8%, respectively.

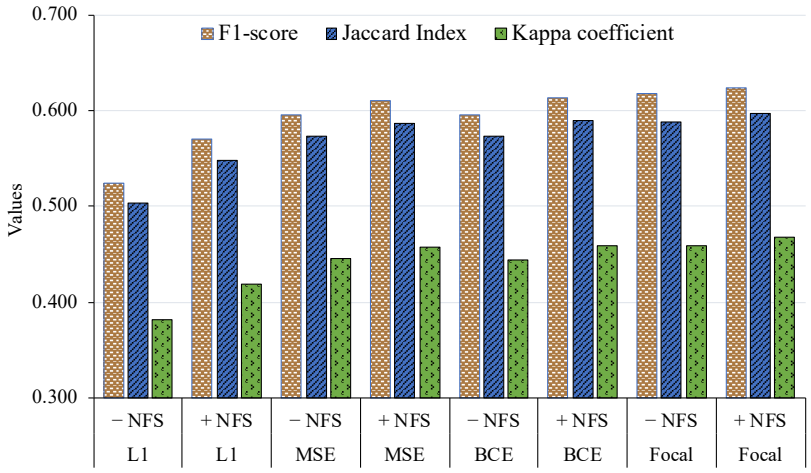
3.3. Qualitative Results

Figure 9 presents six representative results of outlines extracted from the model trained by L1 loss with/without the NFS under test dataset. The backgrounds, red lines, and green circles represent the aerial input, predicted outline, and focused area. In general, the addition of the NFS yields a better building outline extraction, particularly on shadowed areas (e.g., green circles in a, b, and e) and turning corners (e.g., green circles in d and f). Additionally, the model trained with the NFS yields a more intact outline (e.g., green circles in c).

Figure 10 shows six representative groups of building outlines extracted from the model trained by the MSE loss with/without the NFS. Generally, the addition of the NFS yields a slightly better building outline extraction. Using the NFS, the extracted outlines contain fewer false positives within buildings (e.g., green circles in a and b) and fewer breakpoints (e.g., green circles c, d, e, and f).

Figure 11 shows six representative groups of outlines extracted from the model trained by BCE loss with or without the NFS. The backgrounds, red lines, and green circles represent the aerial input, predicted outline, and focused area, respectively. As shown in the figure, the addition of the NFS yields a slightly better line extraction at areas shadowed by surrounding trees (e.g., green circles of column a, e, and f). Moreover, the additional NFS results in better line continuity around corners of the buildings

(e.g., green circles of column b, c, and d). In general, using the proposed NFS, the building outline extracted from the aerial image is more intact, particularly on building corners and shadowed areas.



(a) Bar chart

Loss	Condition	F1-score	Jaccard Index	Kappa coefficient
L1	− NFS	0.524	0.503	0.382
L1	+ NFS	**0.571**	**0.548**	**0.419**
MSE	− NFS	0.596	0.573	0.445
MSE	+ NFS	**0.611**	**0.587**	**0.458**
BCE	− NFS	0.596	0.573	0.444
BCE	+ NFS	**0.613**	**0.589**	**0.459**
Focal	− NFS	0.618	0.588	0.459
Focal	+ NFS	**0.624**	**0.597**	**0.468**

(b) Table

Figure 8. Performances of different losses, either with or without nearest feature selector (NFS). (**a**) Bar chart for comparison of relative performances (**b**) Table of performances under different loss functions. For each loss function, the highest values are highlighted in bold.

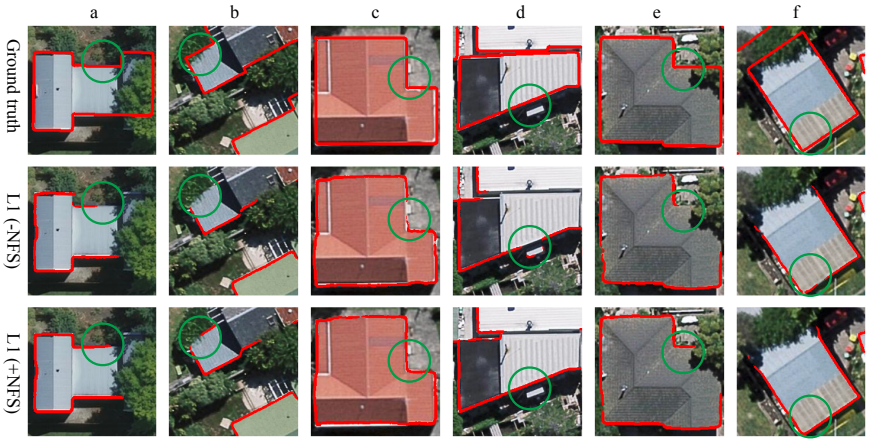


Figure 9. Representative results of extracted outlines from model trained by L1 loss with/without nearest feature selector (NFS). Backgrounds, red lines, and green circles represent aerial input, predicted outline, and focused area, respectively. Selected results are denoted as (**a**–**f**).

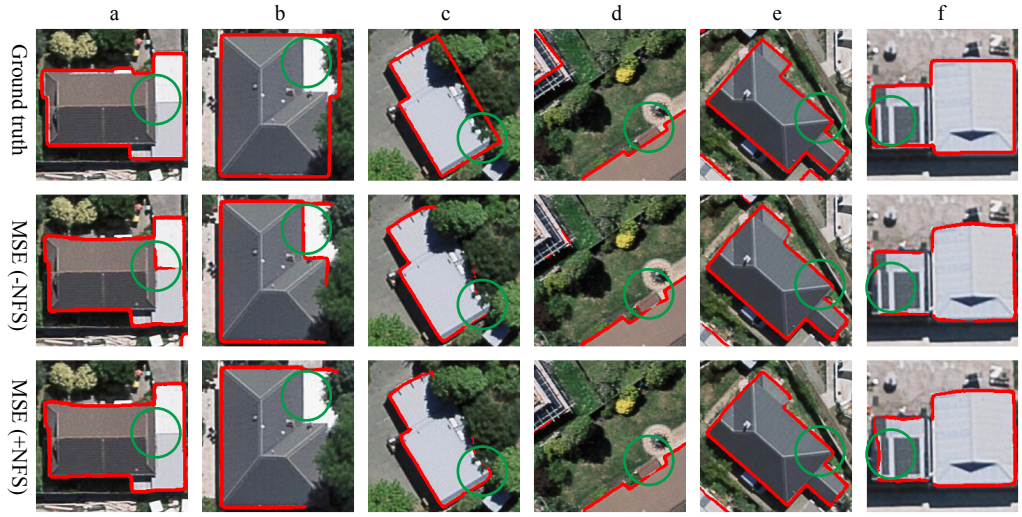


Figure 10. Representative results of outlines extracted from model trained by mean square error (MSE) loss with/without nearest feature selector (NFS). Backgrounds, red lines, and green circles represent aerial input, predicted outline, and focused area, respectively.Selected results are denoted as (**a**–**f**).

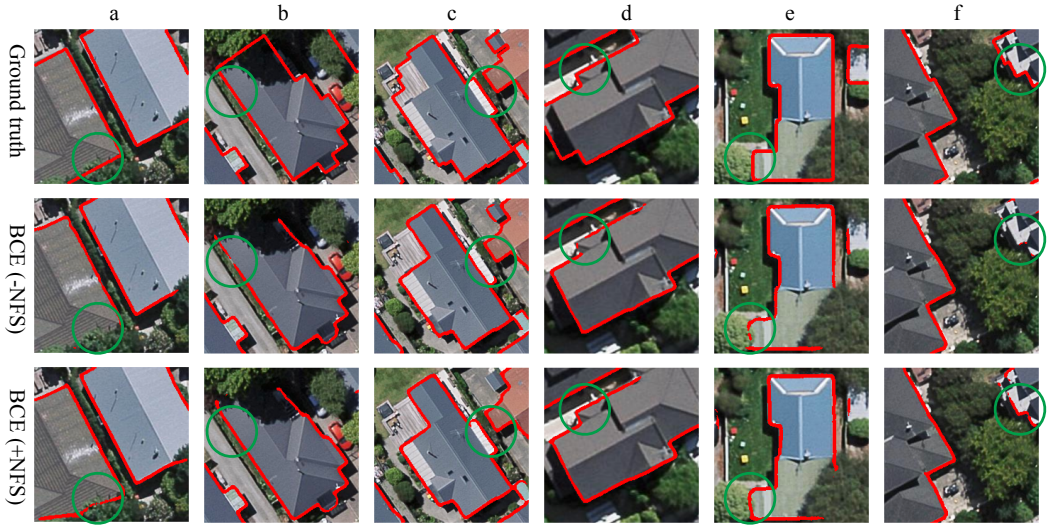


Figure 11. Representative results of outlines extracted from model trained by binary cross-entropy (BCE) loss with/without nearest feature selector (NFS). Backgrounds, red lines, and green circles represent aerial input, predicted outline, and focused area, respectively.Selected results are denoted as (**a**–**f**).

Figure 12 presents six representative pairs of building outlines extracted from the model trained with the focal loss with or without the NFS. Owing to the robustness of the focal loss, even without the NFS, the model successfully recognizes and extracts the major parts of the building outline from the aerial input (e.g., b, c, and f). However, with the additional NFS, the generated outlines contain fewer false positives around corners with complicated backgrounds (e.g., a, d and e). Compared with L1 loss, the addition of NFS imposes a less significant effect on the model trained with focal loss. This observation is consistent with the quantitative result shown in Figure 8b.

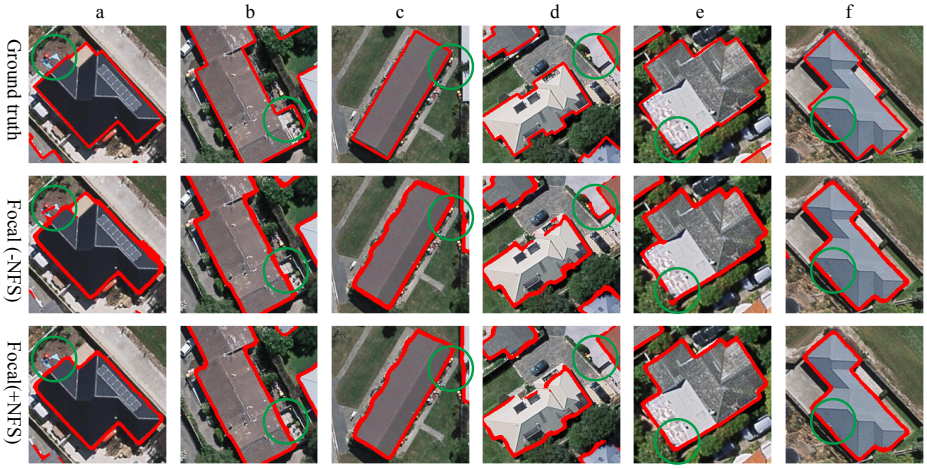


Figure 12. Representative results of outlines extracted from model trained by focal loss with/without nearest feature selector (NFS). Backgrounds, red lines, and green circles represent aerial input, predicted outline, and focused area, respectively.Selected results are denoted as (**a**–**f**).

Figure 13 presents four representative pairs of failure cases from the model trained with the loss function that combines with or without the nearest feature selector (NFS). As compared with the model trained without NFS, the addition of NFS might lead to un-expected misclassification around corners.

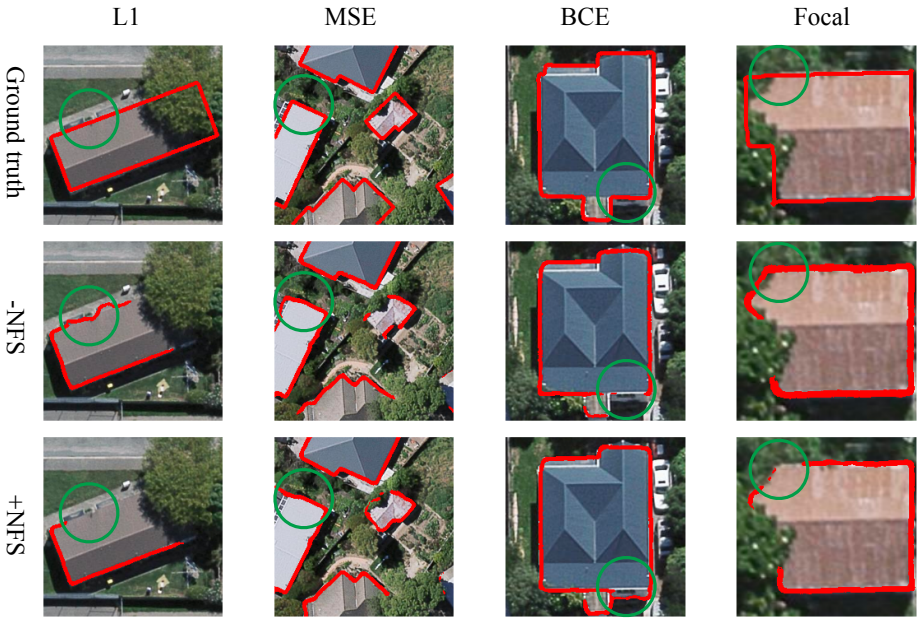


Figure 13. Representative failure cases of outlines extracted from model trained by four losses with/without nearest feature selector (NFS). Backgrounds, red lines, and green circles represent aerial input, predicted outline, and focused area, respectively.

3.4. Computational Efficiency

All experiments are trained and tested on a Sakura "koukakuryoku" Server (https://www.sakura.ad.jp/koukaryoku/) equipped with a 4*times* NVIDIA Tesla V100 GPU (https://www.nvidia.com/en-us/data-center/tesla-v100/) and installed with 64-bit Ubuntu 16.04 LTS. The original SegNet is implemented on Caffe [50] and trained on multi-class scene segmentation tasks, CamVid road scene segmentation [51] and SUN RGB-D indoor scene segmentation [52]. The stochastic gradient descent (SGD) with a fixed learning rate of 0.1 and a momentum of 0.9 is applied to train the

model. The implementation of the modified SegNet is based on geoseg (https://github.com/huster-wgm/geoseg) [53], which is built on top of Pytorch(version $\geq$ 0.4.1). To avoid interference by other hyperparameters, all models are trained with a fixed batch size (i.e., 24) and a constant iteration (i.e., 10,000). The Adam stochastic optimizer, which operates at default settings (lr = 2^{-4}, betas = [0.9, 0.999]), is used for training different models.

Table 1 shows the computing speeds of the methods in frames per second (FPS). Among all the loss functions, the additional NFS results in slightly longer processing time during both training and testing. However, the decline in PFS is not significant.

Table 1. Comparison of the computational efficiencies of different loss functions under conditions that with or without NFS.

Loss	Condition	Training FPS	Testing FPS
L1	−NFS	102.3	264.4
L1	+NFS	98.5	236.1
MSE	−NFS	101.9	265.9
MSE	+NFS	98.4	236.2
BCE	−NFS	102.1	266.8
BCE	+NFS	98.7	236.6
Focal	−NFS	101.6	268.5
Focal	+NFS	97.9	236.3

4. Discussion

4.1. Regarding the NFS

In recent years, fully convolutional networks have demonstrated their ability in automatically extracting line features, including roads and building outlines [36,39,54]. However, those studies mainly focused on designing deeper or more complex network architectures to enhance the representation capability for better predictions. The loss functions of fully convolutional networks cannot handle misalignments or rotations between inputs and manually created annotations. Because the building outline occupies a small portion of pixels, misalignments and rotations will severely interfere with the building outline extraction accuracy.

Herein, we propose the NFS module to dynamically re-align the prediction and corresponding annotation. The proposed framework can be easily appended into existing loss functions, such as L1, MSE, and focal loss. Through a dynamic re-alignment, the addition of NFS enables the correct position of the annotation to be located for an appropriate loss calculation. Qualitative and quantitative results based on the testing data demonstrated the effectiveness of our proposed NFS.

4.2. Accuracies, Uncertainties, and Limitations

Among all methods, the focal loss with NFS indicates the highest values for all evaluation metrics. Its values of the f1-score, Jaccard index, and kappa coefficient are 0.624, 0.597, and 0.468. Compared with the naive L1 loss, the addition of the NFS results in significant increments in all evaluation metrics. The increments of the f1-score, kappa coefficient, and Jaccard index reach 8.8%, 8.9%, and 9.8%, respectively. As it is arguable that the kappa coefficient is unsuitable in the assessment and comparison of the accuracy [55], the actual performance gained from the NFS might be less significant (i.e., less than 9.8%). For robust loss functions (e.g., focal, and BCE loss), the improvement afforded by the NFS is less significant (see details in Figure 8b). Owing to the sliding-and-matching mechanism, the proposed NFS cannot be applied to annotations that require rotation correction. Since the methods are designed and trained on image patches with dense buildings, the trained model is not appropriate for evaluating the entire study area where buildings are sparsely presented.

We observe a slight decrease in processing speed when the NFS is applied through the analysis of computational efficiency. Considering the performance gain by the NFS, computational efficiency degradation is negligible. Because the NFS is independent of the aerial characteristic, in principle, it should apply for not only aerial images, but also other data sources (e.g.satellite, SAR, and UAV). The effectiveness of the NFS will be further estimated using publicly available datasets from various sources [56].

Because of the extremely biased negative/positive ratio, complete building outline extraction is still challenging. With the current classification-based scheme, the model is trained to generate pixel-to-pixel prediction using features extracted from sequential convolutional layers. The predicted pixels of the building outline lack of internal connectivity that some pixels might be misclassified as non-outline (e.g., 2nd and 3rd rows in Figure 9).

5. Conclusions

For an accurate building outline extraction, we design a nearest feature selector (NFS) module to dynamically re-align predictions and slightly misaligned annotations. The proposed module can be easily combined with existing loss functions to manage subpixel or pixel-to-level misalignments of the manually created annotations more effectively. For all loss functions, the addition of the proposed NFS yielded significantly better performances in all the evaluation metrics. For the classic L1 loss, the increments gained by using the additional NFS are 8.8%, 8.9%, and 9.8% for the f1-score, kappa coefficient, and Jaccard index, respectively. We plan to improve the similarity selection mechanism and apply it to other data sources to achieve better generalization capacity for large-scale applications.

Author Contributions: Conceptualization, G.W.; Investigation, Y.W.; Project administration, G.W.; Resources, R.S.; Validation, Y.G. and Y.H.; Writing—original draft, Y.W.; Writing—review & editing, G.W. All authors have read and agreed to the published version of the manuscript.

Funding: Part of this work was supported by the JST (Japan Science and Technology Agency) aXis Grant Number JPMJAS2019.

Acknowledgments: We gratefully acknowledge SAKURA Internet Inc. for the provision of the *koukaryoku* GPU server for our experiments.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

CNN	Convolutional Neural Network
FCN	Fully Convolutional Networks
NFS	Nearest Feature Selector

References

1. Kuffer, M.; Pfeffer, K.; Sliuzas, R. Slums from space—15 years of slum mapping using remote sensing. *Remote Sens.* **2016**, *8*, 455. [CrossRef]
2. Pham, H.M.; Yamaguchi, Y.; Bui, T.Q. A case study on the relation between city planning and urban growth using remote sensing and spatial metrics. *Landsc. Urban Plan.* **2011**, *100*, 223–230. [CrossRef]
3. Ordóñez, J.; Jadraque, E.; Alegre, J.; Martínez, G. Analysis of the photovoltaic solar energy capacity of residential rooftops in Andalusia (Spain). *Renew. Sustain. Energy Rev.* **2010**, *14*, 2122–2130. [CrossRef]
4. Hamre, L.N.; Domaas, S.T.; Austad, I.; Rydgren, K. Land-cover and structural changes in a western Norwegian cultural landscape since 1865, based on an old cadastral map and a field survey. *Landsc. Ecol.* **2007**, *22*, 1563–1574. [CrossRef]
5. Li, M.; Zang, S.; Zhang, B.; Li, S.; Wu, C. A review of remote sensing image classification techniques: The role of spatio-contextual information. *Eur. J. Remote Sens.* **2014**, *47*, 389–411. [CrossRef]
6. Chen, R.; Li, X.; Li, J. Object-based features for house detection from rgb high-resolution images. *Remote Sens.* **2018**, *10*, 451. [CrossRef]

7. Xu, B.; Jiang, W.; Shan, J.; Zhang, J.; Li, L. Investigation on the weighted ransac approaches for building roof plane segmentation from lidar point clouds. *Remote Sens.* **2015**, *8*, 5. [CrossRef]
8. Huang, Y.; Zhuo, L.; Tao, H.; Shi, Q.; Liu, K. A novel building type classification scheme based on integrated LiDAR and high-resolution images. *Remote Sens.* **2017**, *9*, 679. [CrossRef]
9. Gilani, S.A.N.; Awrangjeb, M.; Lu, G. An automatic building extraction and regularisation technique using lidar point cloud data and orthoimage. *Remote Sens.* **2016**, *8*, 258. [CrossRef]
10. Guo, Z.; Wu, G.; Song, X.; Yuan, W.; Chen, Q.; Zhang, H.; Shi, X.; Xu, M.; Xu, Y.; Shibasaki, R.; et al. Super-Resolution Integrated Building Semantic Segmentation for Multi-Source Remote Sensing Imagery. *IEEE Access* **2019**, *7*, 99381–99397. [CrossRef]
11. Sahoo, P.K.; Soltani, S.; Wong, A.K. A survey of thresholding techniques. *Comput. Vis. Graph. Image Process.* **1988**, *41*, 233–260. [CrossRef]
12. Kanopoulos, N.; Vasanthavada, N.; Baker, R.L. Design of an image edge detection filter using the Sobel operator. *IEEE J. Solid-State Circuits* **1988**, *23*, 358–367. [CrossRef]
13. Wu, Z.; Leahy, R. An optimal graph theoretic approach to data clustering: Theory and its application to image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **1993**, *15*, 1101–1113. [CrossRef]
14. Tremeau, A.; Borel, N. A region growing and merging algorithm to color segmentation. *Pattern Recognit.* **1997**, *30*, 1191–1203. [CrossRef]
15. Gómez-Moreno, H.; Maldonado-Bascón, S.; López-Ferreras, F. Edge detection in noisy images using the support vector machines. In *International Work-Conference on Artificial Neural Networks*; Springer: Berlin/Heidelberg, Germany, 2001; pp. 685–692.
16. Zhou, J.; Chan, K.; Chong, V.; Krishnan, S.M. Extraction of brain tumor from MR images using one-class support vector machine. In Proceedings of the 2005 IEEE Engineering in Medicine and Biology 27th Annual Conference, Shanghai, China, 17–18 January 2006; pp. 6411–6414.
17. Xie, S.; Tu, Z. Holistically-nested edge detection. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1395–1403.
18. Viola, P.; Jones, M. Rapid object detection using a boosted cascade of simple features. In Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2001, Kauai, HI, USA, 8–14 December 2001; Volume 1, p. I .
19. Lowe, D.G. Object recognition from local scale-invariant features. In Proceedings of the Seventh IEEE International Conference on Computer Vision, Kerkyra, Greece, 20–27 September 1999; Volume 2, pp. 1150–1157.
20. Ojala, T.; Pietikainen, M.; Maenpaa, T. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Trans. Pattern Anal. Mach. Intell.* **2002**, *24*, 971–987. [CrossRef]
21. Dalal, N.; Triggs, B. Histograms of oriented gradients for human detection. In Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), San Diego, CA, USA, 20–25 June 2005; Volume 1, pp. 886–893.
22. Inglada, J. Automatic recognition of man-made objects in high resolution optical remote sensing images by SVM classification of geometric image features. *ISPRS J. Photogramm. Remote Sens.* **2007**, *62*, 236–248. [CrossRef]
23. Aytekin, Ö.; Zöngür, U.; Halici, U. Texture-based airport runway detection. *IEEE Geosci. Remote Sens. Lett.* **2013**, *10*, 471–475. [CrossRef]
24. Dong, Y.; Du, B.; Zhang, L. Target detection based on random forest metric learning. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2015**, *8*, 1830–1838. [CrossRef]
25. LeCun, Y.; Bengio, Y. Convolutional networks for images, speech, and time series. *Handb. Brain Theory Neural Netw.* **1995**, *3361*, 1995.
26. Ciresan, D.; Giusti, A.; Gambardella, L.M.; Schmidhuber, J. Deep neural networks segment neuronal membranes in electron microscopy images. In *Advances in Neural Information Processing Systems*; Curran Associates: Red Hook, NY, USA, 2012; pp. 2843–2851.
27. Guo, Z.; Shao, X.; Xu, Y.; Miyazaki, H.; Ohira, W.; Shibasaki, R. Identification of village building via Google Earth images and supervised machine learning methods. *Remote Sens.* **2016**, *8*, 271. [CrossRef]
28. Long, J.; Shelhamer, E.; Darrell, T. Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 3431–3440.

29. Kampffmeyer, M.; Salberg, A.B.; Jenssen, R. Semantic segmentation of small objects and modeling of uncertainty in urban remote sensing images using deep convolutional neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 1–9.
30. Badrinarayanan, V.; Kendall, A.; Cipolla, R. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 2481–2495. [CrossRef] [PubMed]
31. Noh, H.; Hong, S.; Han, B. Learning deconvolution network for semantic segmentation. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1520–1528.
32. Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*; Springer: Berlin/Heidelberg, Germany, 2015; pp. 234–241.
33. Lin, T.Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature pyramid networks for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; Volume 1, p. 4.
34. Wu, G.; Shao, X.; Guo, Z.; Chen, Q.; Yuan, W.; Shi, X.; Xu, Y.; Shibasaki, R. Automatic Building Segmentation of Aerial Imagery Using Multi-Constraint Fully Convolutional Networks. *Remote Sens.* **2018**, *10*, 407. [CrossRef]
35. Wu, G.; Guo, Y.; Song, X.; Guo, Z.; Zhang, H.; Shi, X.; Shibasaki, R.; Shao, X. A stacked fully convolutional networks with feature alignment framework for multi-label land-cover segmentation. *Remote Sens.* **2019**, *11*, 1051. [CrossRef]
36. Wu, G.; Guo, Z.; Shi, X.; Chen, Q.; Xu, Y.; Shibasaki, R.; Shao, X. A Boundary Regulated Network for Accurate Roof Segmentation and Outline Extraction. *Remote Sens.* **2018**, *10*, 1195. [CrossRef]
37. Mnih, V.; Hinton, G.E. Learning to detect roads in high-resolution aerial images. In *European Conference on Computer Vision*; Springer: Berlin/Heidelberg, Germany, 2010; pp. 210–223.
38. Wei, Y.; Wang, Z.; Xu, M. Road structure refined CNN for road extraction in aerial image. *IEEE Geosci. Remote Sens. Lett.* **2017**, *14*, 709–713. [CrossRef]
39. Zhang, Z.; Liu, Q.; Wang, Y. Road extraction by deep residual u-net. *IEEE Geosci. Remote Sens. Lett.* **2018**, *15*, 749–753. [CrossRef]
40. Zhou, L.; Zhang, C.; Wu, M. D-LinkNet: LinkNet With Pretrained Encoder and Dilated Convolution for High Resolution Satellite Imagery Road Extraction. In Proceedings of the CVPR Workshops, Salt Lake City, UT, USA, 18–22 June 2018; pp. 182–186.
41. Polak, M.; Zhang, H.; Pi, M. An evaluation metric for image segmentation of multiple objects. *Image Vis. Comput.* **2009**, *27*, 1223–1227. [CrossRef]
42. Carletta, J. Assessing agreement on classification tasks: The kappa statistic. *Comput. Linguist.* **1996**, *22*, 249–254.
43. Li, E.; Femiani, J.; Xu, S.; Zhang, X.; Wonka, P. Robust rooftop extraction from visible band images using higher order CRF. *IEEE Trans. Geosci. Remote Sens.* **2015**, *53*, 4483–4495. [CrossRef]
44. Comer, M.L.; Delp, E.J. Morphological operations for color image processing. *J. Electron. Imaging* **1999**, *8*, 279–290. [CrossRef]
45. Nair, V.; Hinton, G.E. Rectified linear units improve restricted boltzmann machines. In Proceedings of the 27th International Conference on Machine Learning (ICML-10), Haifa, Israel, 21–24 June 2010; pp. 807–814.
46. Ioffe, S.; Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Proceedings of the International Conference on Machine Learning, Lille, France, 6–11 July 2015; pp. 448–456.
47. Nagi, J.; Ducatelle, F.; Di Caro, G.A.; Cireşan, D.; Meier, U.; Giusti, A.; Nagi, F.; Schmidhuber, J.; Gambardella, L.M. Max-pooling convolutional neural networks for vision-based hand gesture recognition. In Proceedings of the 2011 IEEE International Conference on Signal and Image Processing Applications (ICSIPA), Kuala Lumpur, Malaysia, 16–18 November 2011; pp. 342–347.
48. Shore, J.; Johnson, R. Properties of cross-entropy minimization. *IEEE Trans. Inf. Theory* **1981**, *27*, 472–482. [CrossRef]
49. Lin, T.Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal loss for dense object detection. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2980–2988.

50. Jia, Y.; Shelhamer, E.; Donahue, J.; Karayev, S.; Long, J.; Girshick, R.; Guadarrama, S.; Darrell, T. Caffe: Convolutional architecture for fast feature embedding. In Proceedings of the 22nd ACM International Conference on Multimedia, Orlando, FL, USA, 3–7 November 2014; ACM: New York, NY, USA, 2014; pp. 675–678.
51. Brostow, G.J.; Fauqueur, J.; Cipolla, R. Semantic object classes in video: A high-definition ground truth database. *Pattern Recognit. Lett.* **2009**, *30*, 88–97. [CrossRef]
52. Song, S.; Lichtenberg, S.P.; Xiao, J. Sun rgb-d: A rgb-d scene understanding benchmark suite. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 567–576.
53. Wu, G.; Guo, Z.; Shao, X.; Shibasaki, R. Geoseg: A Computer Vision Package for Automatic Building Segmentation and Outline Extraction. In Proceedings of the IGARSS 2019-2019 IEEE International Geoscience and Remote Sensing Symposium, Yokohama, Japan, 28 July–2 August 2019; pp. 158–161.
54. Wu, S.; Du, C.; Chen, H.; Xu, Y.; Guo, N.; Jing, N. Road Extraction from Very High Resolution Images Using Weakly labeled OpenStreetMap Centerline. *ISPRS Int. J. Geo-Inf.* **2019**, *8*, 478. [CrossRef]
55. Foody, G.M. Explaining the unsuitability of the kappa coefficient in the assessment and comparison of the accuracy of thematic maps obtained by image classification. *Remote Sens. Environ.* **2020**, *239*, 111630. [CrossRef]
56. Chen, Q.; Wang, L.; Wu, Y.; Wu, G.; Guo, Z.; Waslander, S.L. Aerial Imagery for Roof Segmentation: A Large-Scale Dataset towards Automatic Mapping of Buildings. *arXiv* **2018**, arXiv:1807.09532.