Building Detection from Satellite Imagery using Ensemble of Size-specific Detectors

Ryuhei Hamaguchi Shuhei Hikosaka
PASCO CORPORATION
{riyhuc2734, saykua3447}@pasco.co.jp

Abstract

In recent years, convolutional neural networks (CNNs) show remarkably high performance in building detection tasks. While much progress has been made, there are two aspects that have not been considered well in the past: how to address a wide variation in building size, and how to well incorporate with context information such as roads. To answer these questions, we propose a simple, but effective multi-task model. The model learns multiple detectors each of which is dedicated to a specific size of buildings. Moreover, the model implicitly utilizes context information by simultaneously training road extraction task along with building detection task. The road extractor is trained by distilling knowledge from another pre-trained CNN, requiring no labels for roads in its training. Our experiments show that the proposed model significantly improves the building detection accuracy.

1. Introduction

Automatic detection of buildings from remote sensing imagery has been a long-standing goal. The task is of great importance because building maps provide basic information for various kinds of applications including marketing, urban management, and popularity estimation. In recent years, convolutional neural networks (CNNs) show remarkably high performance in building detection tasks [1, 7, 9, 10, 12, 14]. While much progress has been made, there are two aspects that are not well considered.

One aspect is the variation in building size. Figure 1 shows a distribution of building sizes in a dataset used in DeepGlobe competition [2]. In the figure, we can see the wide variety in the building size. In most cases, large buildings and small buildings have very different visual appearances (*e.g.*, a large shopping mall and a small house). Furthermore, due to limited spatial resolutions, small buildings would not be the same as larger ones even if up-scaled. Despite such variations, previous works treat all buildings into a single class and do not well consider the variation they have.

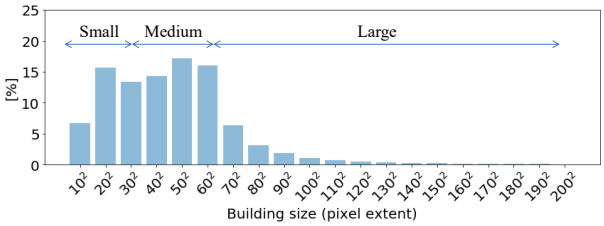

Figure 1. Distribution of building size in the dataset used in DeepGlobe competition [2].

To deal with the variation in this paper, we treat the detection of buildings of different sizes as different tasks. Specifically, we propose a multi-task model that learns size-specific detectors for detecting each size of buildings. The multi-task modeling is suitable for the task. At lower levels, the detectors can share general features for all buildings while at higher levels, they can concentrate on learning specific features for each kind of buildings.

The other aspect is the utilization of context information. The context information around buildings is sometimes helpful. Especially, information about roads helps to recognize buildings because of the co-occurrence between them. Actually, in [12], the accuracy of building detection is improved by training multi-class model which simultaneously detects both of buildings and roads.

One problem of [12] is that to train their multi-class segmentation model, they require training samples that have labels for both classes. The samples that have only labels for either of the classes cannot be used for training, which significantly limits the number of training samples available.

To fully utilize all the labels available, we propose to add a road extraction branch to the proposed multi-task model stated above. The branch is trained by knowledge distillation [6] using another road extraction model as a teacher. By using the output of the teacher model as a ground truth, the road extraction branch can be trained even for samples that have only building labels.

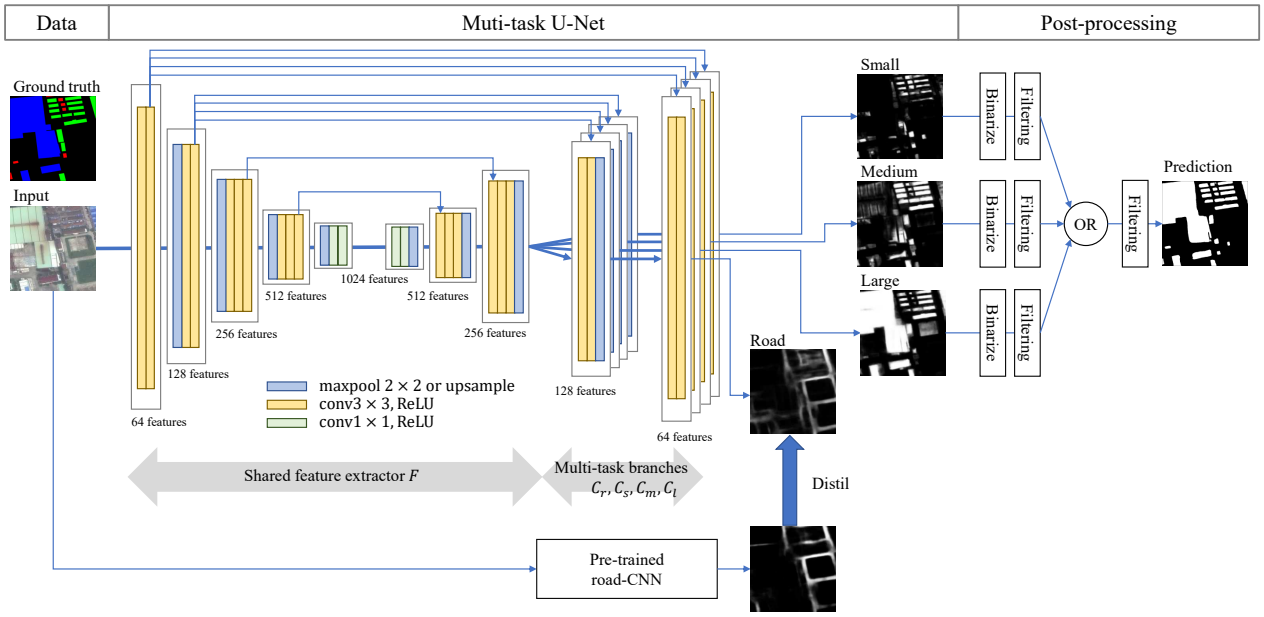

Figure 2. Overview of the proposed method. The multi-task U-Net (Center) consists of a shared feature extractor F and successive multi-task branches C_r, C_s, C_m, and C_l. The model takes RGB images as input and outputs four types of probability maps: one for road extraction result and three for building detection results. Each of the three results corresponds to "small", "medium" and "large" size buildings. The road extraction branch is trained by knowledge distillation from another pre-trained CNN model. The building detection branches are trained using multi-size labels shown on the left side. In the labels, red, green and blue color represent "small", "medium" and "large" buildings respectively. Finally, the outputs for each size of buildings are integrated into a final prediction result in the post-processing (Right)

In our experiments, the proposed multi-task model with the knowledge distillation shows the remarkable performance improvement. All the experiments are conducted through the participation of DeepGlobe competition [2].

2. Methods

2.1. Overview

Figure 2 shows an overview of the proposed model. The model architecture is based on U-Net [11]. The model consists of a shared feature extractor F and successive multi-task branches C_r, C_s, C_m, and C_l. Each of the branches solves different tasks: road extraction (C_r) and size-specific building detection (C_s, C_m, C_l). The loss function is defined as the sum of the losses from each branch.

$$\mathcal{L} = \mathcal{L}_{small} + \mathcal{L}_{medium} + \mathcal{L}_{large} + \mathcal{L}_{road}. \quad (1)$$

Bellow, we explain the size specific building detection branches (section 2.2), the road extraction branch (section 2.3), and the post-processing (section 2.4).

2.2. Size-specific building detection

The proposed model has three building detection branches (C_s, C_m, C_l), each of which is responsible for detecting "small", "medium", and "large" buildings. For input $\boldsymbol{x} \in \boldsymbol{X}$, the outputs of the branches can be written as follows.

$$\boldsymbol{p}^k = C_k(F(\boldsymbol{x})), k = \{s, m, l\}. \quad (2)$$

To train the branches, the multi-class labels $y_i = \{c_n, c_s, c_m, c_l\}$ which defines "non-building", "small", "medium", and "large" classes are used. Note that the multi-class labels can be acquired from commonly used binary building labels. An example of the labels is shown in the left side of Figure 2. Using the labels, the loss function of the branch for "small" buildings becomes

$$\mathcal{L}_{small} = \sum_i I(y_i = c_s) \log p_i^s + I(y_i = c_n) \log (1 - p_i^s). \quad (3)$$

Here, $I(\cdot)$ is an indicator function which returns 1 if the argument is true and returns 0 if false. Note that in the loss function, only the pixels of small buildings or non-buildings affect the loss value. This is because pixels which is out of scope of the branch (*i.e.* "medium" and "large") produces unwanted error signal. The other loss functions ($\mathcal{L}_{medium}$ and $\mathcal{L}_{large}$) are defined in the same way as $\mathcal{L}_{small}$.

2.3. Distillation from road extraction model

The output of the road extraction branch C_r is defined as follows.

$$\boldsymbol{p}^r = C_r(F(\boldsymbol{x})) \quad (4)$$

The branch is trained by distilling knowledge from a teacher model R which is trained in advance using another dataset for road extraction. For each image $\boldsymbol{x} \in \boldsymbol{X}$ in the building detection dataset $\boldsymbol{X}$, the output of the teacher model is calculated as

$$\hat{\boldsymbol{p}} = R(\boldsymbol{x}). \tag{5}$$

During training, the teacher output $\hat{\boldsymbol{p}}$ is used as a ground truth for input $\boldsymbol{x}$. Using cross entropy, the loss function for the branch is defined as follows.

$$\mathcal{L}_{road} = \sum_i \hat{p}_i \log p_i^r + (1 - \hat{p}_i) \log (1 - p_i^r) \tag{6}$$

2.4. Post-processing

In the post-processing, the output probability maps from each building detection branches are integrated. The integration is conducted in the following steps. First, the outputs are binarized using pre-defined threshold. Then, from each binarized maps, invalid building masks that are out of assigned building size are filtered out. Then, filtered results are integrated by taking logical sum for each pixel. Finally, too small building masks are removed as invalid predictions.

To determine the binarization threshold, each of the size-specific branches are evaluated on a validation set. For each branch, the best performing threshold is chosen from range [0.4, 0.6] in terms of recall value calculated on the assigned size of ground truth masks. The range of the filtering is defined by the half of lower bound and the twice of upper bound of the assigned range for each branch. This means that we leave all the prediction masks that are possible to have IoU > 0.5 with any ground truth masks in their assigned range. Although the model performance is not so sensitive to the choice of the filter range, we find that too strict filtering (*e.g.* no margin to the filtering range) or too loose filtering (*e.g.* no filtering) degrades the performance.

3. Experiments

3.1. Dataset

In our experiments, we used building detection dataset provided in DeepGlobe competition [2]. Among the provided images, we used 30cm resolution RGB images (RGB-Pansharpen). For internal validation, we hold out randomly chosen 300 images from 10560 training images. Throughout the paper, we used the validation data to evaluate our models. For training, 5 million patches of size 128×128 are cropped from the training images. When cropping, class balance is considered as far as possible.

To define the division of building sizes, k-means clustering is applied to the square root of the building extent in the dataset. As a result, the division becomes as follows: $S \leq 1150$, $1150 < S \leq 4540$ and $4540 < S$ for "small", "medium" and "large" buildings respectively, where S denotes a pixel extent of a building. In this paper, we only tried $k = 3$, but it would be possible to have more or fewer classes.

Table 1. F1 scores for each model evaluated on the internal validation set and the final phase test set of DeepGlobe competition. For validation set, F1 scores are shown for each size of buildings as well as overall buildings.

	F1 @ val				F1 @ final
Model	small	medium	large	overall	overall
VGG-U-Net	45.07	80.24	80.04	67.4	-
+Aug.	38.48	79.41	79.75	65.36	
+Distil	47.27	81.69	79.83	68.95	-
+SS	53.91	85.11	84.83	72.36	-
+SS+Distil	53.65	85.28	83.49	72.32	-
Ensemble	54.70	85.52	85.12	73.04	71.99
Res-U-Net	50.51	83.97	82.88	71.60	-
+SS	**56.92**	85.89	**86.12**	74.41	73.70
Ensemble	56.90	**86.51**	85.96	**74.67**	**73.91**

Table 2. F1 score for different choice of branch point for multi-task detectors. For each model, the position of branch point is changed from 3rd block to 5th block in the decoder (*i.e.*, lower to higher layer)

	Branch point		
Model	3rd block	4th block	5th block
VGG-U-Net+Distil	70.31	69.91	70.25
VGG-U-Net+SS	72.18	71.76	69.25

For training of the teacher model for distillation, we used the road extraction dataset provided from the other part of the competition [2].

3.2. Experimental setups

We build two types of U-Net architectures as our baseline: VGG-U-Net and Res-U-Net. Each architecture has the encoder which consists of pool4 features of VGG16 [13] and conv5 features of Resnet-18 [5] respectively, each followed by two 3×3 convolutions with ReLU activation functions. Since the resolution of feature maps are important to detect small buildings [3], we made minor modifications for our Res-U-Net: we eliminated the first max-pooling layer from Resnet-18 and changed stride of conv1 from 2 to 1. As a result, the global stride of the encoder output becomes 16, which is the same as VGG-U-Net. For both of the architectures, the decoders have the symmetric architecture as the encoders. Then, these baselines are extended to the proposed multi-task U-Net by adding multiple branches, *i.e.* the size-specific branches (SS), the road extraction branch (Distil), and the combination of them (SS+Distil). All the branches have identical architecture and they branch off at the fourth block in the decoder (see Figure 2 for VGG-U-Net+SS+Distil).

We also compare the proposed method to multi-scale training, a commonly used approach to deal with objects in various scales. Specifically, during training of VGG-U-Net, we randomly scale input images by the factor of 0.5,

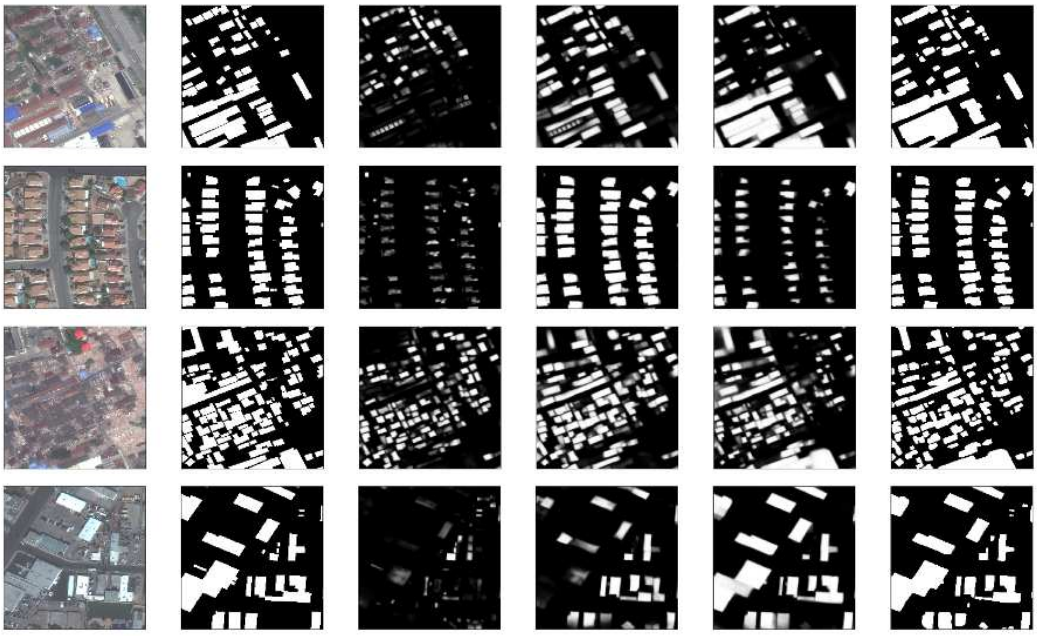

Figure 3. The example results of ensemble model of Res-U-Net family. The output probability maps of size-specific branches are shown through the 3rd to 5th column and final prediction results after post-processing are shown in the last column.

1.0 and 2.0 (VGG-U-Net+Aug.). Although We also tried the augmentation at test time, this significantly hurt the performance because nearby buildings in down-scaled inputs tend to be over segmented and be included in single large masks.

For initialization of the networks, pre-trained weights on ImageNet are used for layers in the encoder and the other layers are initialized using [4]. Adam [8] are used for optimization with an initial learning rate of 1.0e-4 and coefficient of weight decay term of 5.0e-4. The learning rate is decayed linearly along with training iteration.

The teacher model for road extraction has the same architecture and the same training setting as VGG-U-Net.

3.3. Results

Table 1 shows the evaluation results for each model. Compared to the baselines, adding road extraction branch (+Distil) and size specific branches (+SS) improves the performance. The improvement is especially large for "small" buildings. The combination of both branches (+SS+Distil) does not further improve the performance but shows competing performance with size specific branches. The large performance boost is acquired by changing base architecture from VGG16 to Resnet-18.

Table 2 investigate the sensitivity to the choice of the branch point. In the case with size specific branches (+SS), it seems better to branch off early in the decoder, while in the case with road extraction branch (+Distil), the choice of branch point does not affect the performance so much.

3.4. Techniques for performance improvement

All the results in Table 1 are acquired by using test time augmentation. Each of the test images is augmented 6 times with rotation (0, 90, 180 and 270 degrees) and flipping (vertical and horizontal) and the outputs are averaged over the augmented inputs. In addition, we build ensemble models by averaging the output of top-k performing models. We used eight models for VGG-U-Net family and four models for Res-U-Net family. As shown in Table 1, the ensemble of Res-U-Net family performs the best for both of our internal validation score and final test score. Some example outputs of the ensemble model is shown in Figure 3.

4. Conclusion

In this paper, we proposed the multi-task building detection model that can effectively deal with buildings of different size. In addition, the model implicitly utilizes the information about road without using road labels. The proposed model achieved significant improvement compared to the conventional U-Net model. While, in this paper, we decompose the building detection task into subtasks along with the factor of size, there will be some other factors worth exploring, such as the shape complexity or architectural styles. Moreover, there arises a new question: is there a method to automatically define optimal subtask decomposition? We leave these things for future work.

References

[1] Multi-Task Learning for Segmentation of Building Footprints with Deep Neural Networks. *arXiv preprint arXiv:1709.05932*, 2017. 1

[2] 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. *arXiv preprint arXiv:1805.06561*, 2018. 1, 2, 3

[3] R. Hamaguchi, A. Fujita, K. Nemoto, T. Imaizumi, and S. Hikosaka. Effective Use of Dilated Convolutions for Segmenting Small Object Instances in Remote Sensing Imagery. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2018. 3

[4] K. He, X. Zhang, S. Ren, and J. Sun. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. *ICCV*, 2015. 4

[5] K. He, X. Zhang, S. Ren, and J. Sun. Deep Residual Learning for Image Recognition. *CVPR*, 2016. 3

[6] G. Hinton, O. Vinyals, and J. Dean. Distilling the Knowledge in a Neural Network. *NIPS 2014 Deep Learning Workshop*, 2014. 1

[7] V. Iglovikov, S. Mushinskiy, and V. Osin. Satellite Imagery Feature Detection using Deep Convolutional Neural Network: A Kaggle Competition. *arXiv preprint arXiv:1706.06169*, 2017. 1

[8] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *ICLR*, 2015. 4

[9] E. Maggiori, Y. Tarabalka, G. Charpiat, and P. Alliez. Fully convolutional neural networks for remote sensing image classification. *2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, 2016. 1

[10] V. Mnih. *Machine Learning for Aerial Image Labeling*. PhD thesis, 2013. 1

[11] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. *MICCAI*, pages 234–241, 2015. 2

[12] S. Saito, T. Yamashita, and Y. Aoki. Multiple Object Extraction from Aerial Imagery with Convolutional Neural Networks. *Journal of Imaging Science and Technology*, 60, 2016. 1

[13] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, 2014. 3

[14] J. Yuan. Automatic Building Extraction in Aerial Scenes Using Convolutional Networks. *arXiv preprint arXiv:1602.06564*, 2016. 1