Building and road detection from large aerial imagery

Shunta Saito[a] and Yoshimitsu Aoki[a]

[a]Keio University, 3-14-1 Hiyoshi, Kohoku-ku, Yokohama, Kanagawa, Japan

ABSTRACT

Building and road detection from aerial imagery has many applications in a wide range of areas including urban design, real-estate management, and disaster relief. The extracting buildings and roads from aerial imagery has been performed by human experts manually, so that it has been very costly and time-consuming process. Our goal is to develop a system for automatically detecting buildings and roads directly from aerial imagery. Many attempts at automatic aerial imagery interpretation have been proposed in remote sensing literature, but much of early works use local features to classify each pixel or segment to an object label, so that these kind of approach needs some prior knowledge on object appearance or class-conditional distribution of pixel values. Furthermore, some works also need a segmentation step as pre-processing. Therefore, we use Convolutional Neural Networks(CNN) to learn mapping from raw pixel values in aerial imagery to three object labels (buildings, roads, and others), in other words, we generate three-channel maps from raw aerial imagery input. We take a patch-based semantic segmentation approach, so we firstly divide large aerial imagery into small patches and then train the CNN with those patches and corresponding three-channel map patches. Finally, we evaluate our system on a large-scale road and building detection datasets that is publicly available.

Keywords: convolutional neural networks, aerial imagery, semantic segmentation, road detection, building detection

1. INTRODUCTION

Extraction of buildings and roads from aerial imagery has many applications in a wide range of areas including automated map making, urban planning, change detection for real-estate management, land use analysis, and disaster relief. However, these tasks have been performed by human experts manually, so that it is very costly and time-consuming process. Because buildings and roads have much variation in their shape and they may be occluded by other objects such as trees, accurate labeling of large aerial imagery is a complex attentional task for human. Hence, automatic extraction of buildings and roads is highly demanded, and many attempts at automatic aerial imagery interpretation have been proposed in remote sensing literature.

Much of early works use local image features to classify each pixel or segment to an object label, so that how to design the features is critical for the performance of these systems. However, designing robust features for each terrestrial object is difficult because the large number of different objects can be appeared in aerial imagery[1] and they have various appearance. Therefore, there have been some methods to extract multiple features from input images and detect objects using each feature independently from each other, and then finally combine the detection results calculated from different features into one output using decision fusion methods.

Sirmacek et al.[2] proposed a probabilistic framework to detect buildings using four different methods of local feature extraction. Firstly, they separately use Harris corner detector,[3] Gradient-Magnitude-based Support Regions (GMSR),[4] Gabor filtering in different orientations,[5] and Features from Accelerated Segment Test (FAST)[6] to extract features from input images and obtain four different estimation results of candidate object locations. Parameters of these feature extractors are adjusted independently from each other. Then they combine the different estimation results derived from different four features into an integrated building detection output by using data and decision fusion methods.

Further author information: (Send correspondence to Shunta Saito)
Shunta Saito: E-mail: ssaito@aoki-medialab.org, Telephone: +81 (4)5 566 1796
Yoshimitsu Aoki: E-mail: aoki@elec.keio.ac.jp, Telephone: +81 (4)5 566 1796

Image Processing: Machine Vision Applications VIII, edited by Edmund Y. Lam, Kurt S. Niel,
Proc. of SPIE-IS&T Electronic Imaging, SPIE Vol. 9405, 94050K · © 2015 SPIE-IS&T
CCC code: 0277-786X/15/$18 · doi: 10.1117/12.000000

Senaras et al.[7] also proposed a decision fusion method for building detection. They firstly perform mean-shift segmentation to an aerial image with preliminary learned band width parameter, and then calculate Normalized Difference Vegetation Index (NDVI)[8] from red color channel of the aerial image and the corresponding infrared image. The NDVI image is binarized with Otsu's method[9] to extract vegetation segments. They also perform this binarization to a Hue-Saturation-Intensity (HSI) image converted from a three-channel image consists of Infrared-Red-Green and extract shadow regions. Using the result of these pre-processings, both vegetation and shadow segments are excluded from candidate segments. To classify the remaining candidate segments into building or others, they extract fifteen different features from each segment. Then, they train fifteen different classifiers with the features and classify each segment using the classifiers separately and obtain fifteen decisions for each segment. Finally, these fifteen decisions are combined into one decision using Fussy Stacked Generalization.[10]

These decision fusion systems have achieved accurate extraction of terrestrial objects from aerial imagery. However, they have used local image features specially designed for extracting a specific object, and the fusion techniques of multiple classifier decisions have also been intended to be utilized to extract a specific object. There are not as many methods for extracting multiple objects at the same time as for extracting each object separately, though there are many kinds of terrestrial objects in aerial imagery, and the applications (e.g., automated map making) cannot be achieved by extracting only one kind of object, and terrestrial objects may be correlated with each other especially in the case of buildings and roads in urban scenes.

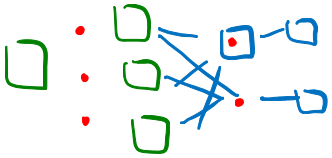
In this paper, we implicitly obtain good feature extractors for both buildings and roads by training Convolutional Neural Networks (CNN) with a publicly available large aerial imagery dataset. We will show that our trained CNN can also classify raw pixels in aerial imagery into building, road or others classes accurately. We never need to design image features manually and train multiple classifiers independently for each terrestrial object. Furthermore, there is no need to consider how to fusion multiple decisions, because the output of the CNN already constructs three-channel estimated label image (Building-Road-Other).

This is not the first work that exploits Neural Networks for aerial imagery interpretation. Mnih et al.[11] proposed a road extraction system using Restricted Boltzmann Machines (RBM). They formulate the problem of extracting road pixels from aerial imagery as a patch-based semantic segmentation. They predict road existence probability distribution from aerial imagery. In their method, firstly an input aerial image is divided into 64×64 patches and applied Principal Component Analysis (PCA) to reduce the dimensionality. Then the PCA vectors are used for fine-tuning of RBM (that has been pre-trained in unsupervised way[12]). They have tested their approach with two datasets that consist of large aerial imagery and binary road label images, and the training set covers roughly 500 square kilometers. This RBM takes a PCA vector of an aerial image patch as input and output a road label image patch. Additionally, to incorporate structures such as road connectivity into the result, they trained a post-processing network that takes the predicted label images as input and output refined label images.

This RBM-based road extraction system has been updated by using CNN and two different probabilistic noise models to consider label noises by Mnih et al.[13] They addressed two types of noises in label images. One is *omission noise* that occurs when an object that appears in an aerial image does not appear in the corresponding label image. Another is *registration noise* that occurs when the location of an object in a label image is inaccurate. They have proposed asymmetric Bernoulli distribution and translational noise distribution to model these noises. They train a CNN with the same dataset, the same problem formulation, and the same forms of input and output (except performing PCA) as the case of their RBM-based approach. Then they finally show the state-of-the-art result of road extraction. However, in a Ph.D. thesis by Mnih,[14] he concluded that label noises have a negative effect, but it is relatively small on predicted results in the case of the system using neural networks.

In this paper, we extend the problem formulation in,[11] the patch-based semantic segmentation of aerial imagery, to consider two different labels simultaneously. Therefore, the output of our system represents not only road labels but also building labels at the same time. These two labels in urban areas are correlated with each other, so considering a trade-off between road and building existence at a single pixel may reduce the confusion between them and improve the performance of prediction.

The rest of this paper is organized as follows. Section 2 presents a brief overview of CNN that is the core model of our system. Section 3 presents the problem formulation to predict road and building labels from aerial



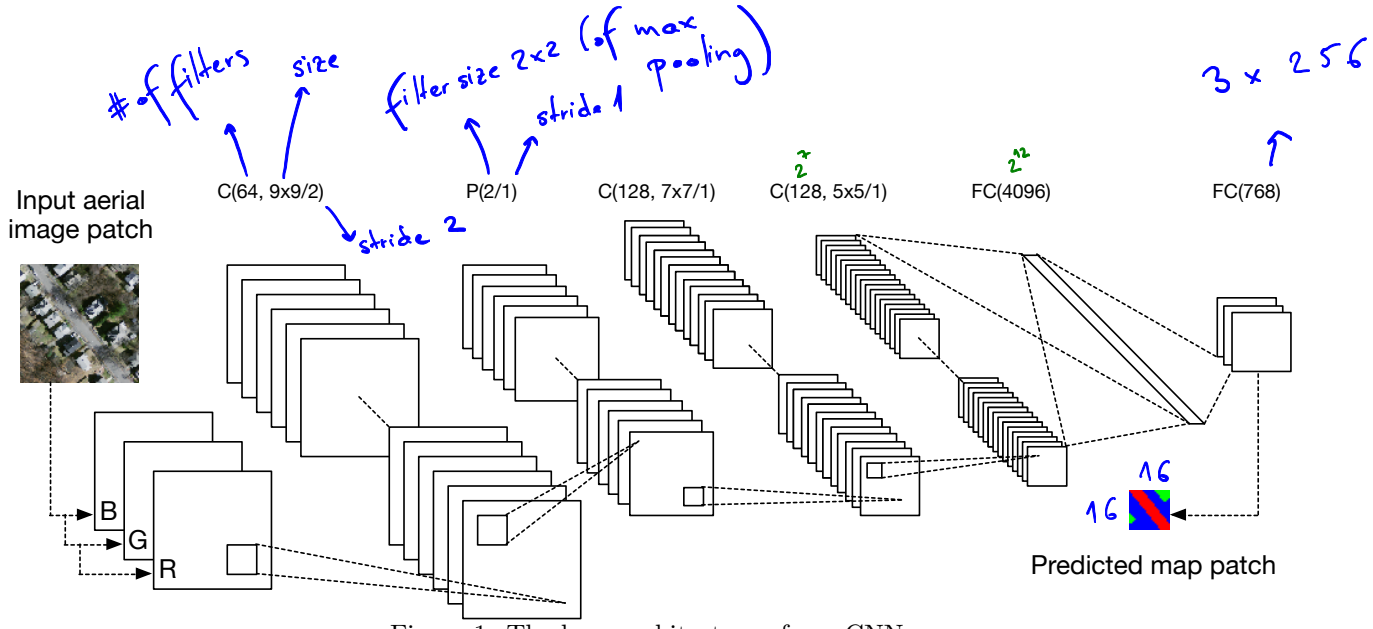


Figure 1: The base architecture of our CNN

imagery simultaneously. Section 4 presents our deep learning framework. Section 5 presents experimental setups and evaluation results of the proposed system under a publicly available datasets proposed by Mnih.[14] Section 6 discusses about the results and summarizes our most important findings.

2. CONVOLUTIONAL NEURAL NETWORKS

In recent years, Convolutional Neural Networks (CNN) have much attention in computer vision area. CNN can be trained as robust feature extractors from raw pixel values and at the same time, learn classifiers for object recognition tasks,[15] regressors for human pose estimation tasks,[16] or mappings for semantic segmentation task.[17] CNN are biologically-inspired variants of multi-layer perceptron. The base idea has been introduced by Fukushima[18] in 1980 as a neural network model for visual recognition tasks. Neocognitron stacks convolutional layers and pooling layers alternately. These layers are inspired by the receptive fields and the hierarchical architecture in cat's visual cortex found by Hubel and Wiesel[19] in 1962, and CNN also have these layers. One of early successful applications of CNN is hand-written digit recognition system proposed by LeCun et al.[20,21] They trained CNN with a classical gradient-based method called backpropagation that has been used to optimize parameters of multi-layer perceptron since Rumelhart et al.[22]

2.1 Base Architecture

The characteristic of CNN is alternatively stacked convolutional layers and spatial polling layers often followed by one or more fully connected layers as in multi-layer perceptron. Fig. 1 shows the base architecture of our CNN. A convolutional layer has a number of filters and convolve them on an input image for extracting features. A pooling layer applies subsampling to the output of the next lower layer for achieving translational invariance.

2.2 Convolutional Layer

A convolutional layer has fixed sized filters. Let K and w be the number of the filters and the size of filters, respectively. A convolutional layer takes a N-channel $(W \times W)$ sized image as input and outputs a K-channel $(W - w + 1 \times W - w + 1)$ sized image. Each channel of this output image is called a filter site. Fig. 2 shows the overview of convolution of filters.

Let y_{ijn}, h_{pqk}, and x_{lmk} be a pixel value at (i, j) on n-th channel of an input image, a weight value at (p, q) on k-th filter, a pixel value at (l, m) on k-th filter site, respectively, x_{lmk} is calculated by

$$x_{lmk} = \sum_{n=1}^{N} \left\{ \sum_{p=0}^{w-1} \sum_{q=0}^{w-1} y_{s \cdot i+p, s \cdot j+q, n} \cdot h_{pqk} \right\} + b_k, \tag{1}$$

where b_k is a bias parameter of k-th filter that is shared among all locations (p, q), so that $b_{pqk} = b_k$. We use tied weights convolutional layers, so that h_{pqk} is shared among all (i, j, n) and it reduces the number of parameters.

Where s is a stride parameter for convolving filters with an interval. If $s > 1$, filters are convolved at intervals of s, so that the size of all filter sites is decreased to $((W - w + 1)/s \times (W - w + 1)/s)$.

Then, activation function f is applied to the resulting filter sites x_{lmk}. Therefore, the output of a convolutional layer is

$$\tilde{y}_{lmk} = f(x_{lmk}). \tag{2}$$

While sigmoid function $f(x_{lmk}) = 1/(1 + \exp(-x_{lmk}))$ is known as a classical activation function for neural networks, we use $f(x_{lmk}) = \max(x_{lmk}, 0)$ and maxout[23] in this paper. The units that perform the former activation function are called rectified linear units (ReLU) and the effectiveness for convergence performance and learning speed is reported in.[24] We also test maxout activation to evaluate the effectiveness for labeling task.

In a convolutional layer, h_{pqk} and b_k are learnable parameters, so we optimize them in training stage of CNN. In the following part of this paper, we describe a convolutional layer that has K filters with the size of $w \times w$ and stride s by $C(K \times w \times w/s)$.

2.3 Pooling Layer

A pooling layer takes filter sites of a convolutional layer and performs subsampling to them. We use max pooling in all pooling layers in this paper. Fig. 3 shows the overview of max pooling operation. Let $\tilde{y}'_{l'm'k}$ be a pixel value at (l', m') on k-th subsampled filter site, it is calculated by

$$\tilde{y}'_{l'm'k} = \max_{0 \le l' \le w'-1, 0 \le m' \le w'-1} y_{s' \cdot l + l', s \cdot m + m', k}. \tag{3}$$

Max pooling layer choose the max value in $(w' \times w')$ sized receptive field, and this operation is applied at intervals of s'. Accordingly, the input K-channel $((W - w + 1)/s \times (W - w + 1)/s)$ sized image is downscaled to the size of $((W - w + 1)/(s \cdot s') \times (W - w + 1)/(s \cdot s'))$. Note that when $s' = 1$, pooling layer does not change the size of input.

In a max pooling layer, there is no learnable parameter. In the following part of this paper, we describe a pooling layer with $(w' \times w')$ sized receptive field and stride s' by $P(w'/s')$.

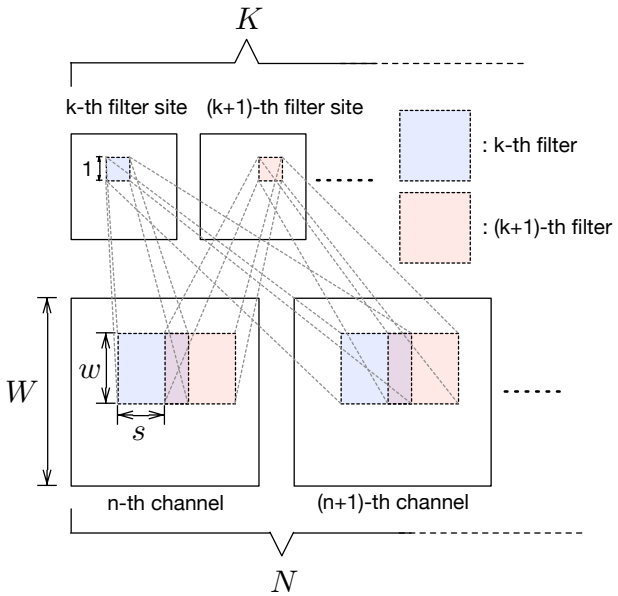


Figure 2: The overview of convolution of filters

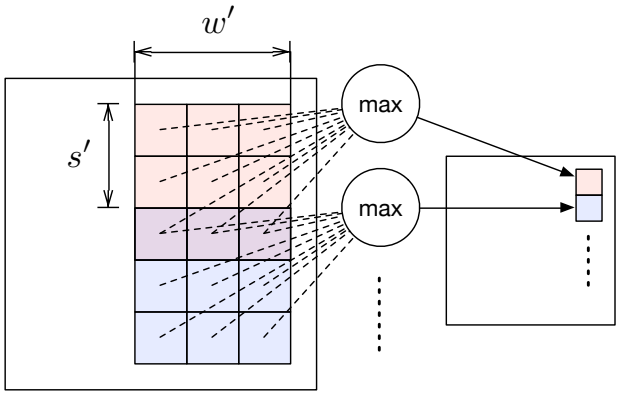


Figure 3: The overview of max pooling operation

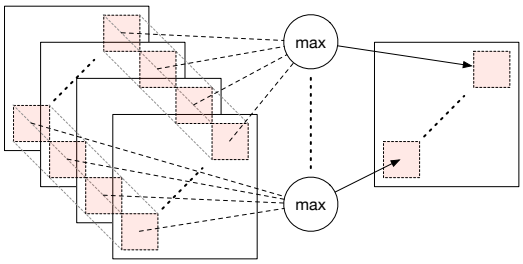


Figure 4: The left image shows an example of input aerial image $\mathbf{S}$, and the right image depicts the corresponding label image $\tilde{\mathbf{M}}$.

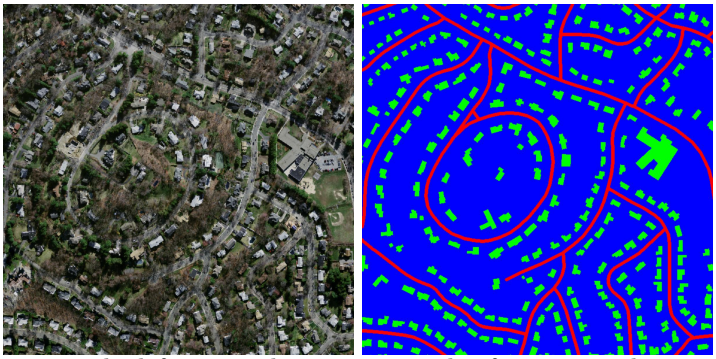


Figure 5: The left image shows an example of input aerial image $\mathbf{S}$, and the right image depicts the corresponding label image $\tilde{\mathbf{M}}$.

2.4 Fully-connected Layer and Dropout

All units in a fully-connected layer are connected to all units in the next lower layer. We describe a fully-connected layer with n_u units by $FC(n_u)$. As shown in Fig. 1, our CNN has two full connections between different layers. One is between $FC(4096)$ and the next lower pooling layer. Another is between $FC(4096)$ and $FC(768)$. During training, the half of connections in the former are randomly chosen and dropped off. This is called Dropout,[25] a technic to prevent overfitting. When inferencing using trained CNN, all the weights in Dropout layers are used halved.

2.5 Maxout Layer

Maxout[23] is an activation function can be substituted for sigmoid and ReLU. The characteristic of maxout is maximum operation over multiple channels of input. Fig. 4 shows the case of maximum operation across 4 feature sites. Maxout performs pooling across channels and reduce the number of filter sites while max pooling reduces the size of filter sites.

In the following part of this paper, we describe a maxout layer pooling across k filter sites by $MO(k)$. This reduces the number of filter sites to $1/k$.

3. PROBLEM FORMULATION

The goal is to predict a multi-channel label image $\tilde{\mathbf{M}}$ from an input aerial image $\mathbf{S}$(Fig. 5). We directly learn a mapping from raw pixels in $\mathbf{S}$ to label image $\tilde{\mathbf{M}}$ with CNN. Let C be the number of object classes of interest, label image $\tilde{\mathbf{M}}$ has $C+1$ channels. Because, it is difficult to consider all kinds of object that can be appeared in aerial imagery, we assign a pixel that belongs not to any of C classes to *otherwise* class. In this paper, we extract two kinds of object, road and building, so that a label image consist of three-channels: road, building, and otherwise. Therefore, let K be the number of all classes including otherwise class, each pixel in a label image is K-dimensional vector. Fig. 6 shows a frame format of label image.

Let $\tilde{\mathbf{M}}_i = [\tilde{M}_{i1}, \ldots, \tilde{M}_{iK}]^{\mathrm{T}}$ be the K-dimensional vector at location i, the elements in this vector have a constraint described as

$$\sum_{k=1}^{K} \tilde{M}_{ik} = 1. \tag{4}$$

This is 1-of-K coding and enables us to consider the trade-off between different classes. Furthermore, each pixel can be considered as a categorical distribution.

In this paper, we formulate the problem by a similar way as what has been proposed by Mnih et al.,[11] so that we actually train CNN to predict a label image patch $n(\tilde{\mathbf{M}}, i, w_m)$ from an aerial image patch $n(\mathbf{S}, i, w_s)$, where $n(\mathbf{I}, i, w)$ denotes the $w \times w$ sized patch of image $\mathbf{I}$ centered at pixel i. To simplify this, we represent a label patch and an aerial image patch as $\tilde{\mathbf{m}}$ and $\mathbf{s}$, respectively. We also assume that pixels in a label patch

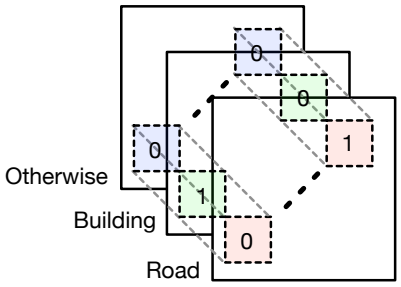


Figure 6: A part of a label image that has three channels. Each pixel is 1-of-K coded label vector.

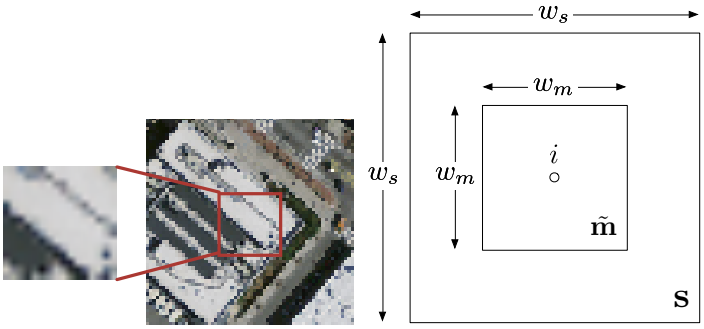


Figure 7: The size difference between label image patch $\tilde{\mathbf{m}}$ and aerial image patch $\mathbf{s}$

are conditionally independent each other given the correspondent aerial image patch. Therefore, a label patch is represented as below conditional distribution:

$$p(\tilde{\mathbf{m}}|\mathbf{s}) = \prod_{i=1}^{w_m^2} p(\tilde{\mathbf{m}}_i|\mathbf{s}), \tag{5}$$

where w_m is the size of a label patch, and it is set to be smaller than the size of an aerial image patch w_s. Fig. 7 shows about this size difference. Both the left two patches show examples of aerial image patch. If an input to CNN has the same size as a predicted label patch as depicted in the leftmost patch in this figure, the CNN have to predict the labels of all pixels in the patch with little context information. However, if CNN takes a larger sized patch such as the second left patch as input, CNN can use some context to predict the labels. This formulation is also proposed by Mnih et al.[11]

4. DEEP LEARNING FRAMEWORK

We propose a deep convolutional neural network that has the base architecture as depicted in Fig. 1. More precisely, our network consists of $C(64, 9 \times 9/1)$ - $MO(4)$ - $P(2/1)$ - $C(128, 7 \times 7/1)$ - $MO(4)$ - $C(128, 5 \times 5/1)$ - $MO(4)$ - $FC(4096)$ - $FC(768)$. The last fully-connected layer is reshaped to $16 \times 16 \times 3$ to calculate the loss defined as below.

4.1 Loss function

Let $w_m \times w_m \times K$ be the form of reshaped output of the CNN, and $\mathbf{x}_i = [x_{i1}, \ldots, x_{iK}]^{\mathrm{T}}$ denote the i-th pixel in this output, softmax operation is applied to each $\mathbf{x}_i$ to convert it into a label probability vector $\hat{\mathbf{m}}_i = [\hat{m}_{i1}, \ldots, \hat{m}_{iK}]^{\mathrm{T}}$. This is calculated as below:

$$\hat{m}_{ik} = \frac{\exp(x_{ik})}{\sum_j \exp(x_{ij})}. \tag{6}$$

Then we define a loss function $\mathcal{L}$ with the summation of pixel-wise cross entropy between a predicted map patch pixel $\hat{\mathbf{m}}_i$ and a label map patch pixel $\tilde{\mathbf{m}}_i$:

$$\mathcal{L} = -\sum_{i=1}^{w_m^2} \sum_{k=1}^{K} \tilde{m}_{ik} \ln \hat{m}_{ik}, \tag{7}$$

where $\tilde{m}_{ik}$ is a k-th element of i-th pixel in a label map patch and can take 0 or 1, so that each term in the above inner summation should be $\ln \hat{m}_{ik}$ (if $\tilde{m}_{ik} = 1$) or 0 (if $\tilde{m}_{ik} = 0$). Then, we calculate the average loss over all $w_m \times w_m \times K$ values by multiplying $\mathcal{L}$ with $1/(w_m^2 \cdot K)$ to calculate errors for backpropagation. Averaging loss enables to choose hyper parameters such as learning rate without considering the label patch size and the number of its channels.

4.2 Learning

We learn all the parameters in CNN by minimizing cross entropy of the training data by using mini-batch stochastic gradient descent with momentum. During learning, we reduce the learning rate by multiplying a fixed reducing rate every τ iterations. Furthermore, we regularize the network using L2 weight decay. Therefore, all hyperparameters in the learning stage are mini-batch size, learning rate (LR) η, LR reducing rate γ, LR reducing frequency τ, a weight of the momentum term α, and a weight of the L2 weight decay β.

Let θ_t be an parameter of CNN at iteration t, the update value at this time is calculated as

$$\Delta\theta_t = \alpha\Delta\theta_{t-1} - \eta_t \left(\frac{\partial \mathcal{L}}{\partial \theta_t} + \beta\theta_t \right), \tag{8}$$

where $\Delta\theta_{t-1}$ denotes a previous update value of the parameter θ.

5. RESULTS

5.1 Dataset

We merged Massachusetts Building Dataset (Mass. Buildings) and Massachusetts Road Dataset (Mass. Roads) for simultaneous extraction of buildings and roads. Both of these datasets are proposed by Mnih[14] and publicly available*. The size of all images in these datasets is 1500×1500, and the resolution is 1m^2/pixel. The building dataset consists of 137 sets of aerial images and corresponding single-channel label images for training part, 10 for testing part, and 4 for validation part. The road dataset consists of 1108 sets for training part, 49 for testing part, and 14 for validation part.

Firstly, we collect aerial images that have both building and road labels from Mass. Buildings and Mass. Roads. Then we found that the all aerial images in the building dataset are included in the road dataset. Therefore, we create a new dataset consists of the same aerial images in the building dataset and synthesize the label images that have three-channels of road, building, and otherwise by stacking road and building label images as different channels. A resulting label image has road label image as R-channel, building label image as G-channel, XOR of these two object channels as B-channel. Examples of an aerial image and the corresponding label image in this dataset are shown in Fig. 5. We call this dataset Massachusetts Road and Building dataset (Mass. RB). The entire dataset covers roughly 340 square kilometers. All images in Mass. RB are completely same as in Mass. Buildings even in terms of training/test/validation subsets, and they are also included in Mass. Roads. Therefore, we can directly compare the performance of our CNN to the results reported by Mnih[14] using Mass. RB dataset.

*http://www.cs.toronto.edu/~vmnih/data/

Table 1: Precision at breakeven point

Model	Road	Building	Others
Mnih-NN[14]	0.8873	0.9150	N/A
Mnih-NN-CRF[14]	0.8904	**0.9211**	N/A
Mnih-NN-MLP[14]	**0.9006**	0.9203	N/A
Plain-Mnih-NN-Multi-ReLU	0.8407	0.8624	0.9738
Plain-Mnih-NN-Multi-ReLU with Dropout	0.8674	0.8994	0.9790
Plain-Mnih-NN-Multi-Maxout with Dropout	0.8586	0.8906	0.9782
Plain-Mnih-NN-Multi-S-ReLU	0.8707	0.8980	0.9799
Plain-Mnih-NN-Multi-S-ReLU with Dropout	0.8843	0.9216	0.9834
Plain-Mnih-NN-Multi-S-Maxout with Dropout	**0.8866**	**0.9230**	0.9834

5.2 Preprocessing

We perform simple preprocessing and data augmentation to the input aerial image patches. All pixel values in each patch is normalized by subtracting the mean value computed over each patch and dividing by the standard deviation computed over the entire dataset. We further rotate each data and label patch with a random angle before passing them to the first layer of our CNN. Random rotation is applied every iteration during learning. Therefore, the completely same pixel intensities may not be inputted to the CNN multiple times because the rotation angle for a input patch is changed every time of feed-forwarding.

5.3 Evaluation Metric

The most common metrics for evaluating road and building detection results are precision and recall. In the remote sensing literature, these are also called correctness and completeness.[26] The precision is the ratio of true road or building pixels to detected pixels as belonging to road or building in predicted map images, while the recall is the ratio of the detected pixels to the true pixels.

However, to compare with the results reported by Mnih,[14] we use the same metric to evaluate our results. They used *relaxed* precision and recall scores instead of exact ones for all experiments. The relaxed precision is defined as the fraction of detected pixels that are within ρ pixels of a true pixel, while the relaxed recall is defined as the fraction of true pixels that are within ρ pixels of a detected pixel. Relaxing the precision and recall in this manner is also used in.[26] In all experiments in this paper, the slack parameter ρ is set to 3 that is the same value used in.[14]

A precision and recall curve consists of 256 precision and recall values at different thresholds. In the other words, to draw the curve, each point is calculated as a set of precision and recall values at threshold t, and t is changed over $[0, 1]$ at even intervals. Then, we summarize this curve with a precision at breakeven point. At a breakeven point, precision and recall values are equal. All values in Table. 1 are precision at breakeven point.

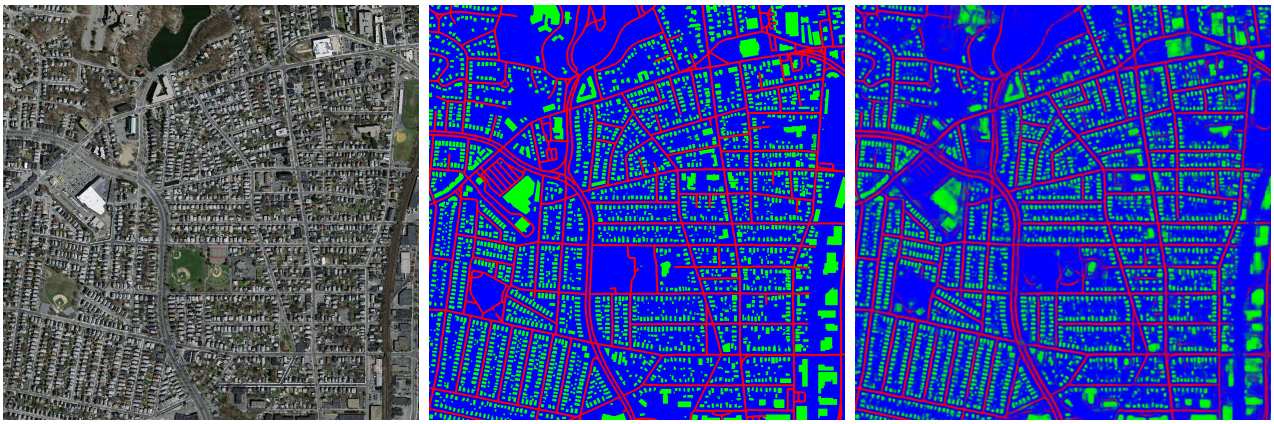
5.4 Models

Table. 1 shows the results reported in[14] and our models. All models proposed by Mnih[14] are learned and evaluated on Mass. Roads and Mass. Buildings separately because all of their models are designed for a single channel prediction. Their base CNN architecture consists of $C(64, 16 \times 16/4)$ - $P(2/1)$ - $C(112, 4 \times 4/1)$ - $C(80, 3 \times 3/1)$ - $FP(4096)$ - $FP(256)$ followed by two noise models. They have considered the label noises with two probabilistic models to improve the prediction result of the base CNN. The result of the base CNN with noise models is shown in Table. 1 as Mnih-NN.

In,[14] other two types of post-processing networks are also proposed to further improve the results of Mnih-NN by considering dependencies between nearby pixels across patches with Conditional Random Fields (CRF) and three-layer perceptrons (MLP). These post-processing networks with CRF and MLP take a 64×64 sized label image patch of the output of Mnih-NN as input and have learned with the same label images used for learning Mnih-NN. The results are shown in Table. 1 as Mnih-NN-CRF and Mnih-NN-MLP.

We firstly evaluate Mnih-NN without the noise models in our problem formulation for multi label prediction. The CNN architecture in Mnih-NN is originally designed for a single label prediction, but in our problem formulation, the output of a network should be able to be reshaped into $16 \times 16 \times 3$, so that the last layer must have 768 units. Therefore, we changes the last layer of Mnih-NN from $FP(256)$ to $FP(768)$ and evaluate this version of the model on our Mass. RB dataset with and without Dropout. These results are shown in Table. 1 as Plain-Mnih-NN-Multi and Plain-Mnih-NN-Multi with Dropout, respectively ("Plain" means that it doesn't include any noise models). We also tested a variant of Plain-Minh-Nn-Multi with Dropout that has Maxout instead of ReLU.

In our models, we also do not consider the label noises and all our models have no post-processing networks. The base model of ours as mentioned in Section 4 is a variant of Plain-Mnih-NN-Multii-Maxout with Dropout. The size of filters in convolutional layers are different. In Table. 1, the result of the base model is shown as Plain-Mnih-NN-Multi-S-Maxout with Dropout. We also tested two variants of this model. One that replaces the all activation functions with ReLU is shown as Plain-Mnih-NN-Multi-S-ReLU and, in addition, another uses Dropout in folly-connected layers. The differences of our models and Mnih's models are the size of filters in



(a) Aerial image (b) Label image (c) Predicted label image

Figure 8: A result of label image prediction by our system

all convolutional layers and whether Maxout and Dropout are used or not, so we call our models by adding 'S' (intend small filters) to the model names.

We have implemented our models with Caffe,[27] a deep learning library.

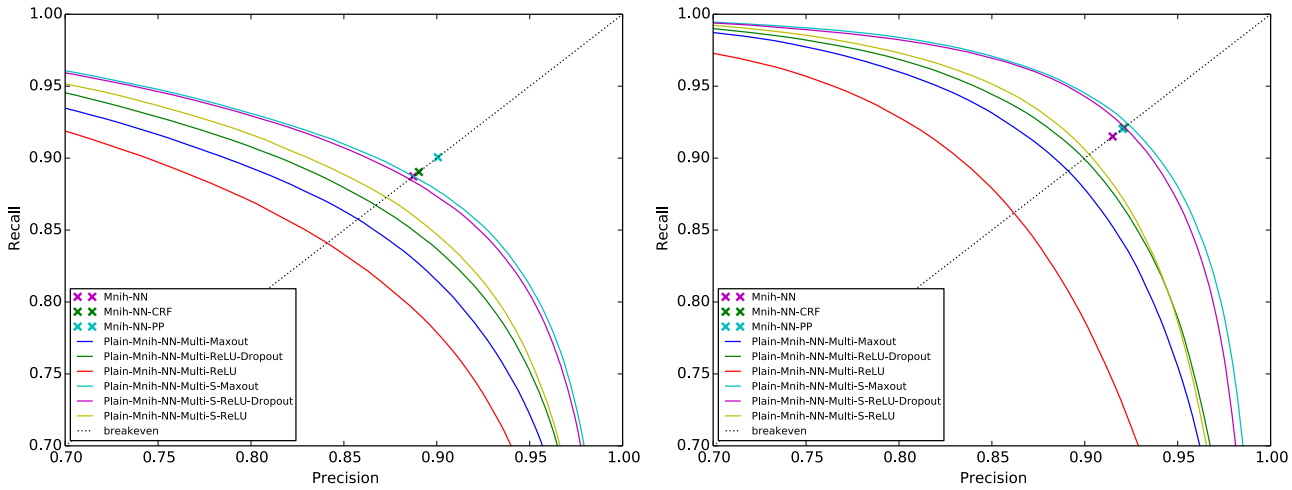
6. DISCUSSION

Firstly, as shown in Table. 1, Dropout improves all the result of label prediction. Secondly, Maxout is better than ReLU at our problem formulation. Both of these technics are regularization and intended to prevent overfitting, and the effectiveness still exists in labeling task such as road and building extraction from aerial imagery.

In Mnih-NN-CRF and Mnih-NN-MLP, only ReLU has been used as activations. Then, CRF and MLP as post-processing network to consider consecutiveness of nearby pixels have been used to improve the result of Mnih-NN. However, in the case of building extraction, the improvement by these additional pre-processings is relatively small than the case of road extraction. Because the Mass. Buildings dataset have many small buildings such as houses but a few large buildings, so that considering larger context information is not important to the performance in the dataset. On the other side, our Plain-Mnih-NN-Multi-S-Maxout with Dropout shows the better result in the case of building extraction. Therefore, the result of our model shows that the better architecture is still enough to improve the result of Mnih-NN in the dataset.

Although the number of road label images in our Mass. RB dataset is eighth part of Mass. Roads dataset, the difference between the result of road channel of Plain-Mnih-NN-Multi-S-Maxout and the result of Mnih-NN on Mass. Roads dataset is 0.0007 (Table. 1). The larger dataset generally improves the generalization ability of deep learning framework, but our model shows about the same performance in road label prediction with significantly small dataset. Combined with the result of building channel, these may be caused by considering the trade-off between the road and building existence and some technics to prevent overfitting such as Maxout, Dropout, and decreasing the size of filters in convolutional layers.

As mentiond in,[14] max-pooling that reduces the size of filters may not lead to improvements because in labeling task, the translational variance is important information to predict labels. To confirm that, we evaluated some models with $P(2/2)$ layers but the result was slightly worse than models with a single $P(2/1)$ after the first convolutional layers. More precisely, we tested a model that consists of $C(64, 9 \times 9/1)$ - $P(2/2)$ - $C(128, 7 \times 7, 1)$ - $P(2/2)$ - $C(128, 5 \times 5, 1)$ - $P(2/2)$ - $FC(4096)$ - $FC(768)$ with Maxout as activations, but the precision at breakeven point of road, building, and otherwise are 0.8799, 0.9171, and 0.9816, respectively. We also tested the same architecture with ReLU and ReLU with Dropout, but the results are little different from the case of the model with Maxout. All precision of these models were small compared with Plain-Mnih-NN-Multi-S-Maxout with Dropout. Therefore, we confirmed that max-pooling with stride $s > 1$ hurts the performance of labeling prediction at least in our dataset and problem formulation.



(a) Precision-recall curve of building label prediction (b) Precision-recall curve of road label prediction

Figure 9: Precision-recall curves

Fig. 9 shows the precision-recall curves of the results of road and building label prediction. In the road prediction, the precision at breakeven points of our models are located in the lower position on the breakeven line that showed as dashed line than the all results of Mnih-NN, Mnih-NN-CRF, and Mnih-NN-PP, especially Mnih-NN-PP is obviously better than the others. This means that considering context over the region that is lager than the size of input patch is important in the case of road prediction. We show an example from the results of prediction by our system in Fig. 8, and a small region extracted from it that includes false negatives because of ignoring consecutiveness in road prediction is shown in Fig. 10. On the other hand, in the building prediction, we achieved state-of-the-art result with Plain-Mnih-NN-Multi-S-Maxout model. This means that more sophisticated model averaging technics such as Maxout and Dropout is more effective to the performance than considering context in our dataset.

Fig. 11 shows the loss values and error rates during learning of Plain-Mnih-NN-Multi-S-Maxout, where an error rate is the breakeven precision of the model evaluated over 12800 patches in test dataset. In this figure, the loss curves and error curves have obviously different shape from each other. This may mean that the loss definition (Eq. 7) is not the best way to represent the difference between predicted patch and label patch. Furthermore, the loss curve did not reach to 0 asymptotically, so that the model architecture and the hyperparameters may still have the room for improvement. The hyperparameters used in this paper were basically shared among all models except learning rate reducing rate and reducing frequency. Mini-batch size was 128. Learning rate was 0.15. The weight of momentum was 0.9. The weight for L2 decay was 0.0002. Learning rate reducing rate was 0.9 for Plain-Mnih-NN-Multi-ReLU and 0.1 for the others. Reducing learning rate was performed every 1000 mini-batch evaluations for Plain-Mnih-NN-Multi-ReLU, 20000 for Plain-Mnih-NN-Multi-S-ReLU-Dropout, 5000 for Plain-Mnih-NN-Multi-S-ReLU, and 30000 for the others.

7. CONCLUSION AND FUTURE WORK

We showed that our multi-channel label prediction models using CNN achieve the equivalent or better performance than the state-of-the-art proposed by Mnih[14] in both road and building detection. Our models predict road labels and building labels simultaneously from an aerial image by predicting three-channel label patch. The trade-off between road and building existence at a pixel is formulated by introducing otherwise channel to predicted label image. In the result of road label prediction, although we have learned our model with one-eighth smaller dataset than the dataset used in,[14] our model have shown about the same precision at breakeven point. In the result of building label prediction, our best model achieve the better performance than Mnih.[14]

We tested six models in our evaluation experiment. The best model was Plain-Mnih-NN-Multi-S-Maxout with Dropout that has smaller filters than filters in the architecture proposed in,[14] Maxout as activations, and

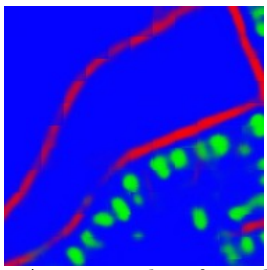


Figure 10: An example of predicted road chunks that ignore the consecutiveness over patches

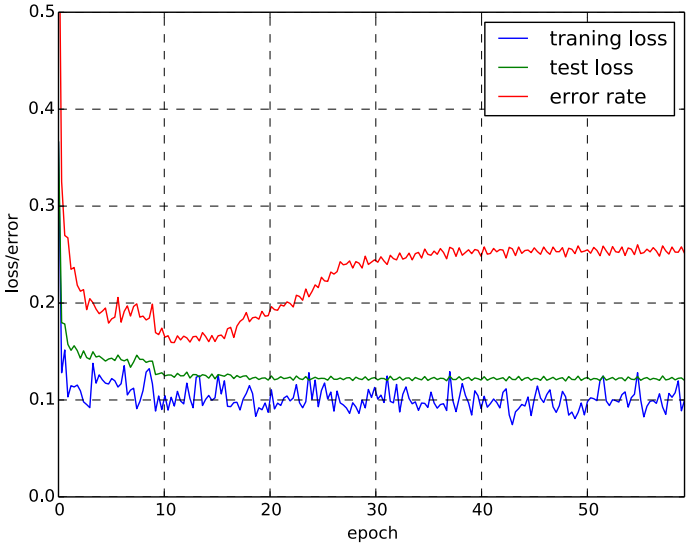


Figure 11: Loss values and error rates that are calculated as precision at breakeven point over some minibatches in test dataset during learning

Dropout used during learning. Therefore, we showed the effectiveness of Maxout and Dropout to aerial imagery labeling task.

Finally, as shown in Fig. 9, there is the room for improvements especially in loss function because decreasing loss value does not always reduce the error rate as shown in Fig. 11. We may further improve the performance of CNN by introducing new loss function that directly maximize area under the precision-recall curve.

REFERENCES

[1] Mayer, H., "Automatic object extraction from aerial imagerya survey focusing on buildings," *Computer Vision and Image Understanding* **74**(2), 138 – 149 (1999).

[2] Sirmacek, B. and Unsalan, C., "A probabilistic framework to detect buildings in aerial and satellite images," *Geoscience and Remote Sensing, IEEE Transactions on* **49**(1), 211–221 (2011).

[3] Harris, C. and Stephens, M., "A combined corner and edge detector.," in [*Alvey vision conference*], **15**, 50, Manchester, UK (1988).

[4] Unsalan, C., "Gradient-magnitude-based support regions in structural land use classification," *Geoscience and Remote Sensing Letters, IEEE* **3**(4), 546–550 (2006).

[5] Vetterli, M. and Kovačević, J., [*Wavelets and subband coding*], vol. 87, Prentice Hall PTR Englewood Cliffs, New Jersey (1995).

[6] Rosten, E., Porter, R., and Drummond, T., "Faster and better: A machine learning approach to corner detection," *Pattern Analysis and Machine Intelligence, IEEE Transactions on* **32**(1), 105–119 (2010).

[7] Senaras, C., Ozay, M., and Yarman Vural, F., "Building detection with decision fusion," *Selected Topics in Applied Earth Observations and Remote Sensing, IEEE Journal of* **6**(3), 1295–1304 (2013).

[8] Tucker, C. J., "Red and photographic infrared linear combinations for monitoring vegetation," *Remote sensing of Environment* **8**(2), 127–150 (1979).

[9] Otsu, N., "A threshold selection method from gray-level histograms," *Automatica* **11**(285-296), 23–27 (1975).

[10] Ozay, M. and Vural, F. T. Y., "A new fuzzy stacked generalization technique and analysis of its performance," *arXiv preprint arXiv:1204.0171* (2012).

[11] Mnih, V. and Hinton, G., "Learning to detect roads in high-resolution aerial images," in [*Proceedings of the 11th European Conference on Computer Vision (ECCV)*], (September 2010).

[12] Hinton, G. E. and Salakhutdinov, R. R., "Reducing the dimensionality of data with neural networks," *Science* **313**(5786), 504–507 (2006).
[13] Mnih, V. and Hinton, G., "Learning to label aerial images from noisy data," in [*Proceedings of the 29th Annual International Conference on Machine Learning (ICML 2012)*], (June 2012).
[14] Mnih, V., *Machine Learning for Aerial Image Labeling*, PhD thesis, University of Toronto (2013).
[15] Krizhevsky, A., Sutskever, I., and Hinton, G. E., "Imagenet classification with deep convolutional neural networks," in [*Advances in neural information processing systems*], 1097–1105 (2012).
[16] Toshev, A. and Szegedy, C., "Deeppose: Human pose estimation via deep neural networks," in [*Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*], 1653–1660 (June 2014).
[17] Farabet, C., Couprie, C., Najman, L., and LeCun, Y., "Learning hierarchical features for scene labeling," *Pattern Analysis and Machine Intelligence, IEEE Transactions on* **35**(8), 1915–1929 (2013).
[18] Fukushima, K., "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position," *Biological cybernetics* **36**(4), 193–202 (1980).
[19] Hubel, D. H. and Wiesel, T. N., "Receptive fields, binocular interaction and functional architecture in the cat's visual cortex," *The Journal of physiology* **160**(1), 106 (1962).
[20] LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., and Jackel, L. D., "Backpropagation applied to handwritten zip code recognition," *Neural computation* **1**(4), 541–551 (1989).
[21] LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P., "Gradient-based learning applied to document recognition," *Proceedings of the IEEE* **86**(11), 2278–2324 (1998).
[22] Rumelhart, D. E., Hinton, G. E., and Williams, R. J., "Learning representations by back-propagating errors," *Cognitive modeling* (1988).
[23] Goodfellow, I. J., Warde-Farley, D., Mirza, M., Courville, A., and Bengio, Y., "Maxout networks," in [*Proceedings of the 30th International Conference on Machine Learning (ICML)*], 1319–1327 (2013).
[24] Nair, V. and Hinton, G. E., "Rectified linear units improve restricted boltzmann machines," in [*Proceedings of the 27th International Conference on Machine Learning (ICML)*], 807–814 (2010).
[25] Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. R., "Improving neural networks by preventing co-adaptation of feature detectors," *arXiv preprint arXiv:1207.0580* (2012).
[26] Wiedemann, C., Heipke, C., Mayer, H., and Jamet, O., "Empirical evaluation of automatically extracted road axes," in [*Empirical Evaluation Techniques in Computer Vision*], 172–187 (1998).
[27] Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., and Darrell, T., "Caffe: Convolutional architecture for fast feature embedding," *arXiv preprint arXiv:1408.5093* (2014).