DOI: 10.2352/ISSN.2470-1173.2016.10.ROBVIS-392
Reprinted from
Journal of Imaging Science and Technology® 60(1): 010402-1–010402-9, 2016.

Multiple Object Extraction from Aerial Imagery with Convolutional Neural Networks

Shunta Saito
Keio University, 3-14-1 Hiyoshi, Kohoku-ku, Yokohama, Kanagawa 223-8522, Japan
E-mail: ssaito@aoki-medialab.org

Takayoshi Yamashita
Chubu University, 1200, Matsumoto-cho, Kasugai-shi, Aichi 487-8501, Japan

Yoshimitsu Aoki
Keio University, 3-14-1 Hiyoshi, Kohoku-ku, Yokohama, Kanagawa 223-8522, Japan

Abstract. *An automatic system to extract terrestrial objects from aerial imagery has many applications in a wide range of areas. However, in general, this task has been performed by human experts manually, so that it is very costly and time consuming. There have been many attempts at automating this task, but many of the existing works are based on class-specific features and classifiers. In this article, the authors propose a convolutional neural network (CNN)-based building and road extraction system. This takes raw pixel values in aerial imagery as input and outputs predicted three-channel label images (building–road–background). Using CNNs, both feature extractors and classifiers are automatically constructed. The authors propose a new technique to train a single CNN efficiently for extracting multiple kinds of objects simultaneously. Finally, they show that the proposed technique improves the prediction performance and surpasses state-of-the-art results tested on a publicly available aerial imagery dataset.*

[DOI: 10.2352/J.ImagingSci.Technol.2016.60.1.010402]

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, disaster relief, etc. However, to date this task has been performed by human experts manually, so that it is a very costly and time-consuming process. Accurate pixel labeling of a large aerial image is a complex attentional task for a human, because terrestrial objects have a great deal of variation in their shapes, and an object may be occluded by other objects such as trees and also by buildings' shadows. Therefore, automatic extraction of buildings and roads is highly demanded and many attempts have been proposed in the remote sensing literature.

There are many previous works that have attempted to extract terrestrial objects from aerial imagery. Many of these works use local image features to classify each pixel or segment to an object label, so that the way in which the features are designed is critical for the performance.

Sirmacek et al.[1] proposed a probabilistic framework to detect buildings, utilizing four different methods for local feature extraction. First, they separately used a Harris corner detector,[2] gradient-magnitude-based support regions (GMSRs),[3] Gabor filtering in different orientations,[4] and a features from accelerated segment test (FAST)[5] to extract feature vectors from aerial imagery and obtain four different estimation results of building locations. The parameters of these methods were adjusted independently for the dataset used in the article. Then they combined the separate estimation results from different features into a single building detection output by utilizing data and decision fusion methods.

Senaras et al.[6] also proposed a decision fusion method for building detection. They first performed mean-shift segmentation to an aerial image with a preliminary learned band width parameter, and then calculated the normalized difference vegetation tndex (NDVI)[7] from the red color channel of the aerial image and the corresponding infrared image. The NDVI image was binarized with Otsu's method[8] to extract vegetation segments. They also performed this binarization on a hue–saturation–intensity (HSI) image converted from a three-channel image consisting of infrared–red–green to extract shadow regions. Using the results of these pre-processings, both vegetation and shadow segments were excluded from the group of all candidate segments. To classify the remaining segments into building class or background, they extracted 15 different image features from each segment. Then, they trained 15 different classifiers with those features and classified each segment by the 15 classifiers independently and obtained 15 decisions for a single segment. Finally, all decisions for a segment were combined by utilizing fussy stacked generalization.[9]

These methods based on decision fusion have achieved accurate extraction of terrestrial objects from aerial imagery. However, they have utilized local image features specially designed for extracting a specific object, and the fusion techniques of multiple decisions have also been intended to

Received June 24, 2015; accepted for publication Nov. 8, 2015; published online Dec. 14, 2015. Associate Editor: Yeong-Ho Ha.

DOI: 10.2352/ISSN.2470-1173.2016.10.ROBVIS-392

extract a specific object. There are not as many methods for extracting multiple objects simultaneously as for extracting each object separately, although there are many different terrestrial objects in aerial imagery, and applications (e.g., automated map making) cannot be achieved by extracting only one object. Furthermore, the occurrence of some terrestrial objects is correlated with other objects, especially in the case of buildings and roads in urban scenes. Therefore, we assume that the consideration of multiple objects simultaneously will improve the performance of extraction of each object.

In this article, we aim at automatically obtaining good feature extractors for both buildings and roads from raw pixel values without any pre-processing. We achieve this by utilizing convolutional neural networks (CNNs) that are trained on a publicly available large aerial imagery dataset.[10] We show that our CNNs can also classify all pixels in aerial imagery into buildings, roads, or background more accurately than previous works.[10–12] Our method does not need to design image features manually, and the training of multiple classifiers independently for each terrestrial object to be extracted is also not needed. Therefore, our method does not need to consider how to fuse multiple decisions, and the output of our CNNs inherently constructs three-channel label images (buildings–roads–background).

The rest of this article is organized as follows. The second section introduces some related works that use CNNs for aerial imagery interpretation, and then presents a brief overview of our contributions. The third section presents a brief overview of convolutional neural networks which constitute the core of our system. The fourth section presents the details of our proposed methods, including the problem formulation to predict building, road and background labels simultaneously. The fifth section presents the training settings of the CNNs. The sixth section describes the datasets that we use for training and evaluation of our proposed methods. The seventh section presents experimental setups and evaluation results of our proposed methods on the datasets. The eighth section discusses the experimental results and the ninth section summarizes our most important findings.

RELATED WORKS AND CONTRIBUTIONS

Approaches utilizing artificial neural networks to solve aerial imagery interpretation problems have already been proposed and have achieved good performance. Mnih et al.[11] proposed a road extraction system utilizing restricted Boltzmann machines (RBMs). They formulated the problem of extraction of road pixels from aerial imagery as a patch-based semantic segmentation task. An input aerial image is divided into 64×64 patches, and principal component analysis (PCA) is applied to them to reduce the dimensionality. Then those PCA vectors are used to fine-tune an RBM that has been pre-trained in an unsupervised way.[13] This RBM predicts a road probability map from a PCA vector of an aerial imagery patch. They finally trained a post-processing network that refines the predicted probability maps to incorporate structures such as road connectivity into the final outputs. They evaluated the performance of their method with a large dataset that consisted of aerial imagery and corresponding binary road label images. The dataset covered roughly 500 km^2.

This RBM-based road extraction method was updated by utilizing convolutional neural networks and incorporating two different noise models.[12] They considered two types of noise occurring in label images, *omission noise* and *registration noise*. The former occurs when an object that appears in aerial imagery does not appear in the corresponding label image. The latter occurs when the location of an object in a label image is inaccurate. They proposed *asymmetric Bernoulli distribution* and *translational noise distribution* to handle these two types of noise. Finally, they showed the effectiveness of the noise models and state-of-the-art results of road extraction. However, in the Ph.D. thesis by Mnih,[10] he concluded that label noise has a negative but relatively small effect on prediction results in the case of the method utilizing neural networks because of its powerful representational ability.

Our contributions in this article are threefold. First, we propose a multi-channel prediction method with a single CNN based on the patch-based formulation of semantic segmentation of aerial imagery.[11] We train a CNN that has a three-channel output layer and predict buildings, roads, and background simultaneously. Second, we propose a new output function for the CNN to consider the specialty of the *background* class. Third, we propose a new model averaging method that can avoid producing patchy results such as those in Figure 6(a).

The first and second proposals are deeply related with each other. The labels that we consider, namely, buildings and roads, are correlated with each other, so we assume that if we exploit the correlation effectively with CNNs, the simultaneous multi-channel prediction will be more accurate compared with predicting each channel independently with different CNNs that have a single-channel output. Thus, we predict the probabilities of both buildings and roads at each pixel. To predict multiple labels, we have to consider the *background* class at the same time. However, there are potentially a large number of classes in the *background* (e.g., tree, river, sea, dog, human, car, etc.), so a feature to represent the *background* class would be difficult to obtain compared with a feature to represent a single class, namely, buildings or roads. Therefore, we assume that if we suppress the effect of the *background* class in the loss function for the training CNNs, the CNNs can learn about building and road representations more effectively. Therefore, we propose a new output function for the CNNs. Utilizing this function during training improves the performance of the CNN that predicts multi-channel label images. Our CNN surpasses state-of-the-art performance; however, it does not need any further machinery such as noise models and structured prediction which are utilized in conventional methods.[10–12]

Our third proposal is a new model averaging technique for semantic segmentation with CNNs. It performs model

DOI: 10.2352/ISSN.2470-1173.2016.10.ROBVIS-392

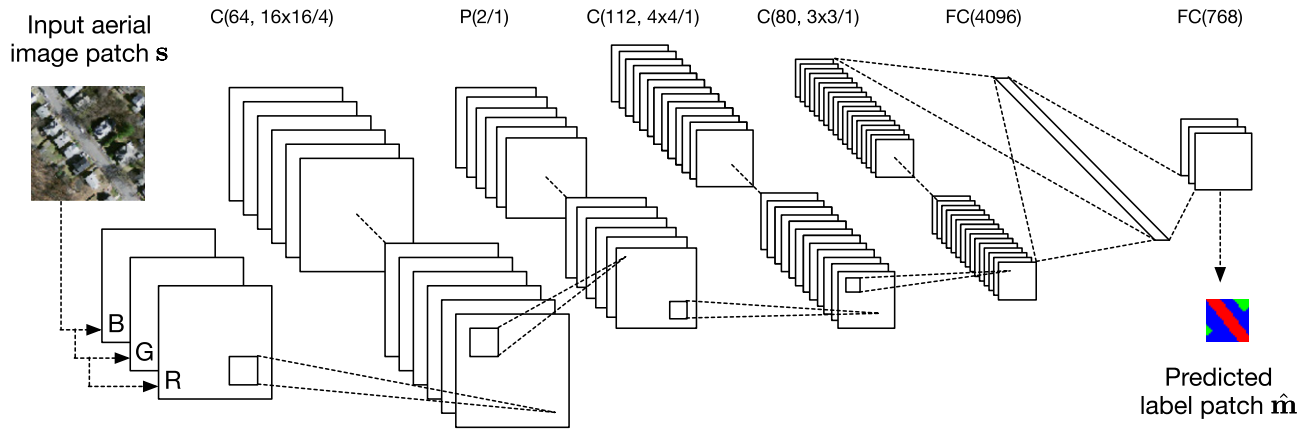


Figure 1. The architecture of the CNN used in this article.

averaging in the conventional sense and also performs smoothing over the outputs simultaneously. This article uses the same CNN architecture as our previous article.[14] However, we substantially improve the performance by utilizing the new output function and this new model averaging technique first proposed in this article.

CONVOLUTIONAL NEURAL NETWORKS

In recent years, convolutional neural networks have attracted much attention in the computer vision area. Convolutional neural networks 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 tasks.[17] A CNN is a biologically inspired variant of a multi-layer perceptron. The base idea of CNNs was introduced by Fukushima[18] in 1980 as a neural network model for visual recognition tasks. The model, Neocognitron, stacks convolutional layers and pooling layers alternately. These layer architectures are inspired by the receptive fields and the hierarchical structure in the cat's visual cortex found by Hubel and Wiesel[19] in 1962. Convolutional neural networks utilize these layer architectures as their components. One of the early successful applications of CNNs is the hand-written digit recognition system proposed by LeCun et al.[20,21] They introduced a way to train CNNs with a classical gradient-based optimization method, backpropagation, which has been utilized to optimize parameters of multi-layer perceptrons since the work of Rumelhart et al.[22]

Base Architecture

The characteristics of CNNs are alternatively stacked convolutional layers and spatial polling layers, often followed by one or more fully connected layers as in a multi-layer perceptron. Figure 1 shows the base architecture we use in this article. A convolutional layer has a number of filters and convolves them on an input image for extracting features. A pooling layer applies subsampling to the output of the next lower layer for acquiring translational invariance. Our CNN has five layers containing trainable parameters. The input is a 64×64-sized three-channel RGB aerial imagery patch, and the output is a 768-dimensional vector, which we reshape to a 16×16-sized three-channel image patch that consists of buildings–roads–background channels.

We describe this architecture as $C(64, 16 \times 16/4)$–$P(2/1)$–$C(112, 4 \times 4/1)$–$C(80, 3 \times 3/1)$–$FC(4096)$–$FC(768)$, where $C(a, b \times b/c)$ means a convolutional layer consisting of a filters that are $b \times b$-sized and the convolution stride is c, $P(a/b)$ means an $a \times a$ max pooling layer with stride b, and $FC(a)$ means a fully connected layer that has a units. We describe this architecture for multi-channel prediction as *ours* in all tables and figures in this article, and the conventional version for single-channel prediction that has 256 units in the last layer[10] is described as *Mnih-CNN*.

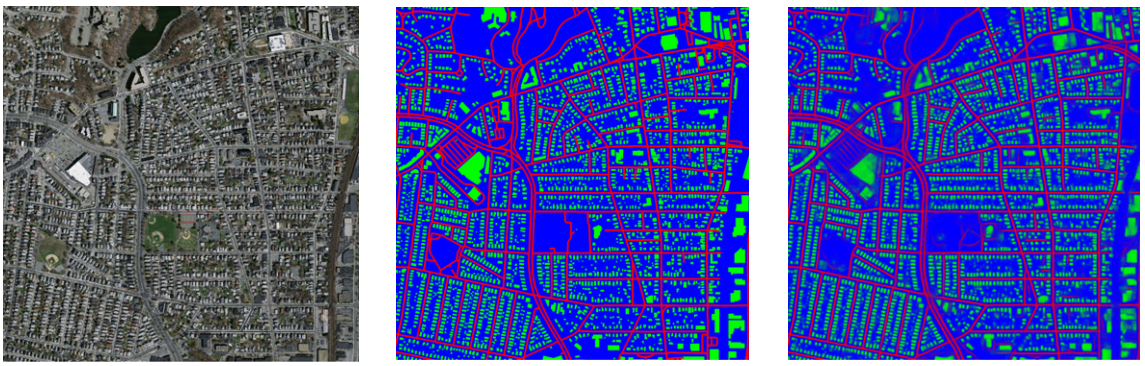
PROPOSED METHOD

The goal is to predict a multi-channel label image $\tilde{\mathbf{M}}$ from an input aerial image $\mathbf{S}$. Figure 2 shows an example of $\mathbf{S}$ and $\tilde{\mathbf{M}}$. We directly learn a mapping from raw pixels in $\mathbf{S}$ to a true label image $\tilde{\mathbf{M}}$ by training the CNN. Let K' be the number of object classes of interest; label image $\tilde{\mathbf{M}}$ has $K' + 1$ channels. Because it is difficult to consider all objects that can appear in aerial imagery as classes of interest, we consider the background class to represent a pixel that does not belong to any of the K' classes of interest. In this article, we extract two objects, buildings and roads, so that a label image consists of three channels, buildings, roads, and background. Let K denote the number of all classes including the background, $K = K' + 1 = 3$. Here, a label image is a K-channel image, so that each single pixel on the label image is a K-dimensional vector. In a label image, the sum over all elements of a pixel vector is always one because each pixel should always be either buildings, roads, or background.

Patch-based Formulation

In this article, we formulate the pixel labeling task in a similar way to what has been proposed by Mnih et al.[10] We train the CNN to predict a $w_m \times w_m$-sized true label patch $\tilde{\mathbf{m}}$ from a $w_s \times w_s$-sized aerial imagery patch $\mathbf{s}$. Each pixel

DOI: 10.2352/ISSN.2470-1173.2016.10.ROBVIS-392



(a) Aerial imagery S (b) Ground truth $\tilde{\mathbf{M}}$ (c) Predicted label image $\hat{\mathbf{M}}$

Figure 2. An example of the resulting predicted label image.

at location i in a true label patch $\tilde{\mathbf{m}}$ is a K-dimensional one-hot vector, $\tilde{\mathbf{m}}_i = [\tilde{m}_{i1}, \tilde{m}_{i2}, \ldots, \tilde{m}_{iK}]$. We describe the output of the CNN by $\hat{\mathbf{m}}$, that is a $w_m \times w_m$-sized predicted label patch. Each pixel at i in a predicted label patch $\hat{\mathbf{m}}$ is also a K-dimensional vector. Here, we assume that all pixels in a true label patch $\tilde{\mathbf{m}}_i$ $(i = 1, \ldots, w_m^2)$ are conditionally independent of each other given a corresponding aerial imagery patch $\mathbf{s}$. Therefore, the posterior of a true label patch given an aerial imagery patch is represented as

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

We train the CNN to maximize this posterior by minimizing the negative log likelihood defined as

$$\mathcal{L} = -\sum_{i=1}^{w_m^2} \ln p(\tilde{\mathbf{m}}_i \mid \mathbf{s}). \tag{2}$$

Figure 3 shows the size difference between the input and output patches. For example, focussing on a small region as depicted in the leftmost patch in Fig. 3, it is difficult to recognize what it is. However, if an input aerial imagery patch has wider region as depicted in the center image in Fig. 3, some context information can be utilized to predict labels, and it can be recognized as a part of a building. Therefore, we set the size of an input patch w_s larger than the size of a predicted label patch w_m, as depicted in the rightmost figure in Fig. 3. This technique for context consideration is also utilized by Mnih et al.[11] and improves the performance.

Channel-wise Inhibited Softmax

In this article, $w_s = 64$, $w_m = 16$, and $K = 3$. We reshape the last layer of the CNN (a 768-dimensional vector) to a $16 \times 16 \times 3$-sized image patch form. Let $\mathbf{x}_i = [x_{i1}, x_{i2}, x_{i3}]^{\mathrm{T}}$ denote the ith pixel in this output patch. The softmax operation

$$\hat{m}_{ik} = \frac{\exp(x_{ik})}{\sum_j \exp(x_{ij})} \tag{3}$$

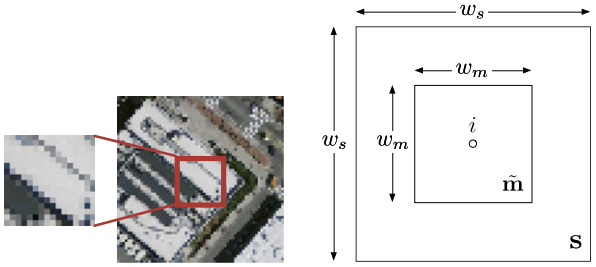


Figure 3. The size difference between the label image patch $\tilde{\mathbf{m}}$ and the aerial imagery patch $\mathbf{s}$.

is applied to each element of $\mathbf{x}_i$ to convert it into a label probability vector $\hat{\mathbf{m}}_i = [\hat{m}_{i1}, \hat{m}_{i2}, \hat{m}_{i3}]^{\mathrm{T}}$. Each posterior at a pixel i is represented as $p(\tilde{\mathbf{m}}_i \mid \mathbf{s}) = \prod_{k=1}^{K} \hat{m}_{ik}^{\tilde{m}_{ik}}$, where $\tilde{\mathbf{m}}_i = [\tilde{m}_{i1}, \tilde{m}_{i2}, \tilde{m}_{i3}]^{\mathrm{T}}$ is a one-hot vector to represent a true label vector at pixel i. The negative log likelihood is calculated as below, and we minimize this loss by stochastic gradient descent[23] with backpropagation:[22]

$$\mathcal{L} = -\sum_{i=1}^{w_m^2} \sum_{k=1}^{K} \tilde{m}_{ik} \ln \hat{m}_{ik}. \tag{4}$$

Here, we propose a new output function called *channel-wise inhibited softmax (CIS)*. We inhibit all units in a specific channel in the output layer of the CNN before calculating softmax. In concrete terms, we put *zero* in all of the output units of the CNN in the background channel, namely, $\forall i, x_{i1}$, and then calculate softmax by Eq. (3). Let $\boldsymbol{\pi}(\cdot)$ denote an element-wise operation of the normal softmax to a vector and $\odot$ denote an element-wise product. The predicted three-channel probability vector at pixel i that is calculated with CIS is

$$\begin{aligned} \hat{\mathbf{m}}_i^{\mathrm{CIS}} &\equiv \boldsymbol{\pi}(\mathbf{c} \odot \mathbf{x}_i), \quad \text{where } \mathbf{c} = [c_1, c_2, \ldots, c_K]^{\mathrm{T}}, \\ \hat{m}_{ik}^{\mathrm{CIS}} &= \frac{\exp(c_k x_{ik})}{\sum_j \exp(c_j x_{ij})}, \quad c_k = \begin{cases} 0, & \text{if } k = 1, \\ 1, & \text{otherwise.} \end{cases} \end{aligned} \tag{5}$$

©2016 Society for Imaging Science and Technology
DOI: 10.2352/ISSN.2470-1173.2016.10.ROBVIS-392

Then the loss function is defined as

$$\mathcal{L}^{\mathrm{CIS}} = -\sum_{i=1}^{w_m^2}\sum_{k=1}^{K} \tilde{m}_{ik} \ln \hat{m}_{ik}^{\mathrm{CIS}}. \tag{6}$$

The derivative of this loss function at x_{ij} is

$$\begin{aligned}\frac{\partial \mathcal{L}^{\mathrm{CIS}}}{\partial x_{ij}} &= -\sum_{k=1}^{K} \tilde{m}_{ik} \frac{1}{\hat{m}_{ik}^{\mathrm{CIS}}} \frac{\partial \hat{m}_{ik}^{\mathrm{CIS}}}{\partial x_{ij}} \\ &= -\sum_{k=1}^{K} c_j \tilde{m}_{ik} (\delta_{jk} - \hat{m}_{ij}^{\mathrm{CIS}}) \\ &= c_j \left\{ \hat{m}_{ij}^{\mathrm{CIS}} \sum_{k=1}^{K} \tilde{m}_{ik} - \sum_{k=1}^{K} \tilde{m}_{ik} \delta_{jk} \right\} \\ &= c_j (\hat{m}_{ij}^{\mathrm{CIS}} - \tilde{m}_{ij}), \end{aligned} \tag{7}$$

where

$$\begin{aligned}\frac{\partial \hat{m}_{ik}^{\mathrm{CIS}}}{\partial x_{ij}} &= \frac{\partial}{\partial x_{ij}} \frac{\exp(c_k x_{ik})}{\sum_{l=1}^{K} \exp(c_l x_{il})} \\ &= \begin{cases} c_j \hat{m}_{ik}^{\mathrm{CIS}} (1 - \hat{m}_{ij}^{\mathrm{CIS}}), & \text{if } j = k, \\ c_j \hat{m}_{ik}^{\mathrm{CIS}} (0 - \hat{m}_{ij}^{\mathrm{CIS}}), & \text{otherwise}, \end{cases} \\ \delta_{jk} &= \begin{cases} 1, & \text{if } j = k, \\ 0, & \text{otherwise}. \end{cases} \end{aligned}$$

It should be noted that $\sum_{k=1}^{K} \tilde{m}_{ik} = 1$ is applied to the last but one line in Eq. (7).

Model Averaging with Spatial Displacement

Model averaging is an ensemble learning method to avoid overfitting. Especially in the context of neural networks, Dropout[24] is known as a way to obtain a similar effect of model averaging within a single network. We utilize Dropout with dropout rate $= 0.5$ in the $FC(4096)$ layer for regularizing networks, but we also explicitly perform model averaging by training multiple CNNs that have the same architecture but are initialized with different weights. In this article, we train each model eight times. Each training stage is started from different initial weights that are sampled from the same Gaussian distribution with different random seeds.

In the inference stage, we give eight displaced input aerial images with different offsets to those eight versions of a single model. We call this operation *model averaging with spatial displacement (MA)*. Figure 4 shows that eight inputs for eight different versions of a single model are displaced by d $(0 \leq d \leq 7)$ pixels from the original image location. The predicted label patches of those versions are tiled with the same displacement d to synthesize a final predicted label image. After tiling those patches, we divide all pixel values by eight for averaging.

LEARNING

We learn all parameters in the CNN by minimizing the loss function with mini-batch stochastic gradient descent with momentum. During learning, we reduce the learning rate by multiplying by a fixed reducing rate every τ iterations. Furthermore, we regularize the network with L2 weight decay. Therefore, the hyper-parameters in the learning stage are the mini-batch size, the learning rate (LR) η, the LR reducing rate γ, the LR reducing frequency τ, a weight of the momentum term α, and a weight of the L2 weight decay β. The learning rate η is started from η_0, and we use the following values for all experiments in this article: $\eta_0 = 0.0005$, $\tau = 10^4$, $\gamma = 0.1$, $\alpha = 0.9$, $\beta = 0.0005$, with the mini-batch size $= 128$. These values are chosen based on Ref. 12.

During training of the CNN, we perform real-time data augmentation to extend the dataset by performing two kinds of transformations to both the input aerial imagery patch and the corresponding true label patch. We apply rotation with a random angle θ and random L–R flip to them. The random angle θ is sampled from the uniform distribution, $\theta \sim U(0, 2\pi)$, for every patch. Both the input aerial imagery patch and the true label patch are rotated around the center point with the same angle. L–R flipping is also performed randomly and equally to both inputs and true label patches.

After performing transformations to the inputs and true labels, the input aerial imagery patch is normalized by subtracting the mean value and dividing by the standard deviation. This procedure is called global contrast normalization (GCN).[25]

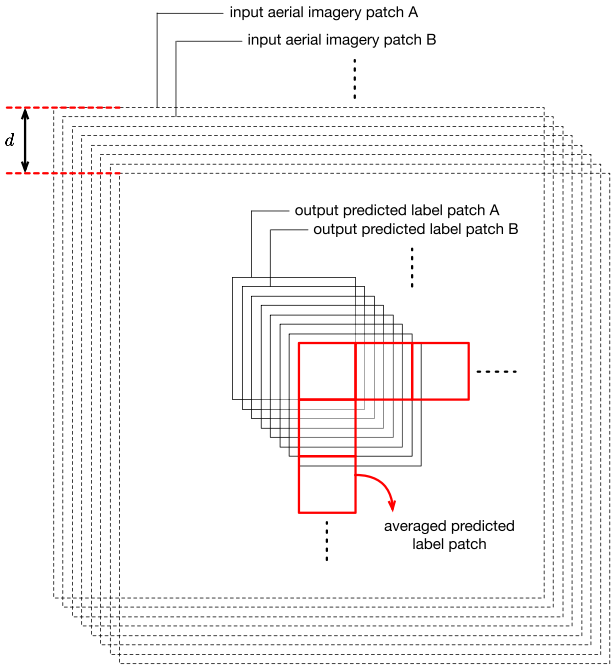


Figure 4. Model averaging with spatial displacement.

DATASET

We built a new dataset by selecting data from two datasets, *Massachusetts Buildings Dataset (Mass. Buildings)* and *Massachusetts Roads Dataset (Mass. Roads)*, which are

©2016 Society for Imaging Science and Technology
DOI: 10.2352/ISSN.2470-1173.2016.10.ROBVIS-392

Table I. Overview of the datasets.

Dataset name	Training	Test	Validation
Massachusetts Roads[10] (Mass. Roads)	1108	49	14
Massachusetts Roads-Mini (*ours*) (Mass. Roads-Mini)	137	10	4
Massachusetts Buildings[10] (Mass. Buildings)	137	10	4
Massachusetts Buildings & Roads (*ours*) (Mass. BR)	137	10	4

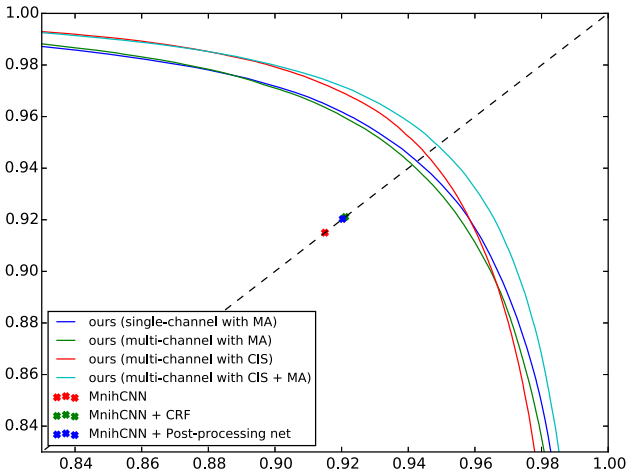
proposed by Mnih[10] and publicly available on website http://www.cs.toronto.edu/~vmnih/data/. We merged *Mass. Buildings* and *Mass. Roads* to create *Massachusetts Buildings and Roads dataset (Mass. BR)* which has multi-channel label images. The size of all images in these datasets is 1500×1500 and the resolution is 1 m^2/pixel. Table I shows the composition of all datasets we use in this article. Both *Mass. Buildings* and *Mass. BR* consist of 151 pairs of aerial images and corresponding single-channel label images. These datasets cover a large area of roughly 340 km^2. Each dataset is divided into three groups. The training, test, and validation subsets of both datasets comprise 137, 10, and 4 images, respectively. On the other hand, *Mass. Roads* consists of 1171 pairs, and these are separated into 1108 training images, 49 test images, and 14 validation images.

To create the *Mass. BR* dataset, we selected aerial images that had both building labels and road labels from *Mass. Buildings* and *Mass. Roads*. Then we found that all aerial images in *Mass. Buildings* are included in *Mass. Roads*. Therefore, we added road labels to *Mass. Buildings*. We synthesized three-channel label images by stacking building, road, and background label images as the three channels. Background labels were created by calculating the XOR of building and road label images. Examples of an aerial image and the corresponding three-channel label image in this new dataset are shown in Fig. 2(b).

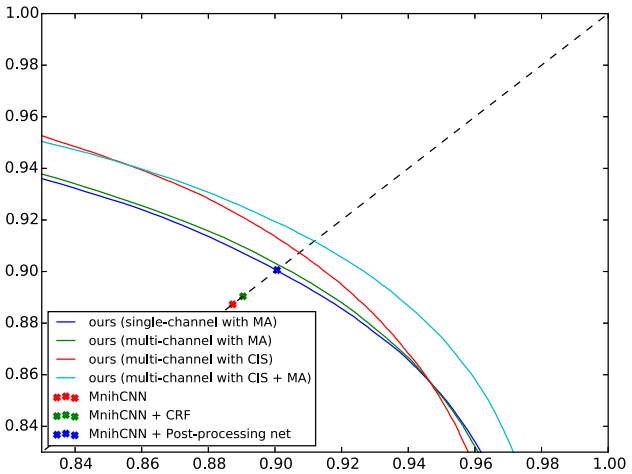
Finally, we found that we cannot compare the result of a model trained on *Mass. BR* with another model trained on *Mass. Roads* directly, because some images in the training set of *Mass. BR* are included in the test set of *Mass. Roads*. Therefore, we took subsets of the *Mass. Roads* dataset and created the *Mass. Roads-Mini* dataset. The training, test, and validation images in *Mass. Roads-Mini* are completely the same as in *Mass. BR*, but *Mass. Roads-Mini* has only road labels. Hence, using *Mass. Roads-Mini* dataset to train a single-channel CNN, we can directly compare the road prediction result with the road *channel* result of the multi-channel CNN.

EXPERIMENT

To show the effectiveness of CIS and MA by comparing with the previous work,[10] we train a single-channel CNN, namely, Mnih-CNN (third section), on *Mass. Buildings* and *Mass. Roads* separately with data augmentation (fourth section) and model averaging with spatial displacement (fourth section).



(a) Precision-recall curve of building label prediction



(b) Precision-recall curve of road label prediction

Figure 5. Precision–recall curves.

Table II shows the performance of the single-channel predictions of the conventional method[12] in the top three rows and the performance of the single-channel prediction with MA. Table III shows each channel's result of *ours*. It should be noted that all of our models are trained with data augmentation, although we do not describe this explicitly in all tables. *Ours (multi-channel with CIS)* means that the model is trained on *Mass. BR* and the loss function is calculated by Eq. (6). All values in these tables are recalls at breakeven points of the *relaxed precision–recall curves* depicted in Figure 5, and the detail is described in the next subsection.

Evaluation Metric

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

DOI: 10.2352/ISSN.2470-1173.2016.10.ROBVIS-392

Table II. Recall at breakeven on the test dataset of the single-channel prediction models.

Model	Mass. Buildings	Mass. Roads	Mass. Road-Mini
Mnih-CNN[10]	0.9150	0.8873	N/A
Mnih-CNN + CRF[10]	0.9211	0.8904	N/A
Mnih-CNN + Post-processing net[10]	0.9203	0.9006	N/A
Ours (single-channel with MA)	**0.9426**	**0.9047**	**0.9005**

Table III. Recall at breakeven on the test dataset of the multi-channel prediction models.

Model	Building channel	Road channel[a]
Ours (multi-channel with MA)	0.9413	0.9019
Ours (multi-channel with CIS)	0.9459	0.9065
Ours (multi-channel with CIS + MA)	**0.9488**	**0.9118**

[a] This can be compared with the results of Road-Mini.

However, to compare our results with the results reported by Mnih,[10] we use the same metric to evaluate our results. They used *relaxed precision and recall scores* instead of exact precision and recall scores 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. Relaxation of the precision and recall in this manner is also used in Wiedemann et al.[26]. Then, in all experiments in this article, the slack parameter ρ is set to 3, which is the same value as used in Wiedemann et al.[26] and Mnih.[10]

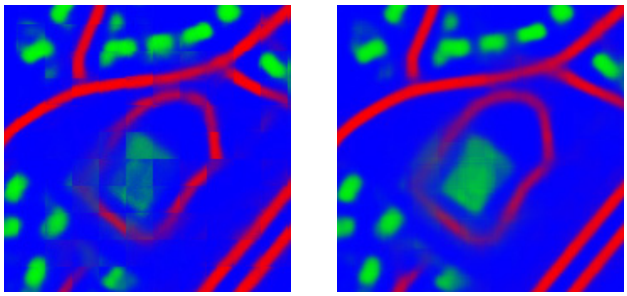
A precision and recall curve consists of many sets of precision and recall values at different thresholds. In other words, to draw the curve, each point is calculated as a set of precision and recall values at threshold t, and t ranges over [0, 1]. Then, we summarize this curve with a recall at the breakeven point. At a breakeven point, the precision and recall values are equal. All values in Tables II–IV are recalls at breakeven points.

Evaluation on Urban Area

To show the benefit of using the correlation between buildings and roads in aerial images, we perform further evaluation. We assume that if our model utilizes the correlation effectively, the performance should be better than the conventional model, with a larger margin in urban regions in which both buildings and roads appear. Because if a patch has only road or building pixels, the correlation cannot be utilized, we extract 64×64-sized patches that have both N_b building pixels and N_r road pixels, where $N_b > w_s^2/K$ and $N_r > w_s^2/K$ from test images. Here, w_m^2 and K are the number of pixels in a patch (64×64) and the number of classes ($K = 3$), respectively. Thus, the extracted patches always include both buildings and roads with a larger area than a third of the whole area of the patch. We evaluate three models on the extracted test patches. Table IV shows the resulting recalls at breakeven points. The row of *ours (single-channel with MA)* shows the results of the single-channel models trained on *Mass. Buildings* or *Mass. Roads-Mini* separately, and MA is used to create the final outputs. These are used to discuss the effectiveness of the multi-channel output by comparing with *ours (multi-channel with MA)* and *ours (multi-channel with CIS + MA)*.

Table IV. Recall at breakeven on the selected region of the test images.

Model	Mass. buildings or buildings channel	Mass. roads-mini or road channel
Ours (single-channel with MA)	0.9418	0.8507
Ours (multi-channel with MA)	0.9539	0.8701
Ours (multi-channel with CIS + MA)	**0.9686**	**0.9020**



(a) without model averaging with spatial displacement

(b) with model averaging with spatial displacement

Figure 6. A prediction result of Mnih-CNN-Multi.

DISCUSSION

For single-channel prediction, Table II shows that *ours (single-channel with MA)* is the best result on all datasets. All the models in this table are to predict a single-channel label image that represents either building or road probability. The conventional methods shown in the top three rows perform noise modeling. Furthermore, *Mnih-CNN + CRF* and *Mnih-CNN + Post-processing net* use conditional random field and additional multi-layer perceptron for structured prediction, respectively. On the other hand, our model does not model label noise specifically and has no post-processing. However, *ours* is more accurate than all of the conventional models on all datasets. The differences between *Mnih-CNN* and *ours (single-channel with MA)* are the use or not of data augmentation during training and MA in the inference stage. Although MA does not perform refinement of results by recognizing any meanings of pixels, MA smoothes the outputs over the boundaries of predicted label patches and improves the performance significantly. This effectiveness is also shown in the case of multi-channel prediction. Fig. 6 shows the effectiveness of MA in a concrete example.

Our multi-channel CNNs output three-channel predicted label patches. Therefore, we extracted each channel to

©2016 Society for Imaging Science and Technology
DOI: 10.2352/ISSN.2470-1173.2016.10.ROBVIS-392

evaluate the accuracy of each class. In Table III, we compared the performance of our proposed techniques on the same base architecture. In this table, *ours (multi-channel with CIS + MA)* achieves the best results in both building and road channels even compared with the results of single-channel predictions. Although the number of training images in *Mass. BR* is one-eighth compared with *Mass. Roads*, the performance of this model trained on *Mass. BR* is better than all of the results of single-channel models trained on *Mass. Roads*. Therefore, we found that *channel-wise inhibited softmax (CIS)*, which assigns zero to all units in the background channel and performs softmax, is a better way to train a CNN for multi-channel semantic segmentation of aerial imagery, and performing MA in the inference stage is also effective.

The reason why CIS improves the performance might be related to the inherent redundancy of the softmax function. In general, subtracting a constant vector from an input vector given to softmax does not affect the output of softmax. For example, let $\mathbf{x}$ be a K-dimensional vector and π denote the softmax function. The ith element of the softmax result of $\mathbf{x} - \mathbf{c}$, where $\mathbf{c} = (c, c, \ldots, c)^{\mathrm{T}}$ (a K-dimensional constant vector), is

$$\begin{aligned}\pi(x_i - c) &= \frac{\exp(x_i - c)}{\sum_{k=1}^{K} \exp(x_k - c)} \\ &= \frac{\exp(-c)\exp(x_i)}{\exp(-c)\sum_{k=1}^{K} \exp(x_k)} = \pi(x_i). \end{aligned} \tag{8}$$

Therefore, there have been some methods to eliminate this redundancy. One of them is to take $x_k \equiv 0$ for some preferred class k, which was introduced by Ripley,[27] and used in Andersen[28] as a *normalized exponential transformation*. Hence, our CIS performs this transformation to each pixel for the multi-channel semantic segmentation problem. Furthermore, we showed that choosing the background class as k in $x_k \equiv 0$ improves the performance in a task that cannot avoid considering "background" or "the others" as one of the objective classes. This is a regularization method for neural networks, so we showed that it is effective even for deep CNNs designed to perform semantic segmentation.

Fig. 5 shows the *relaxed precision–recall curves* of all results. The red, green, and blue cross marks mean the recalls at breakeven points of the conventional models proposed in Mnih.[10] All curves of *ours (multi-channel)* are located above those cross marks in both building and road prediction. As shown in both figures, the results with *CIS* are always better than the results without *CIS*, and the results with *CIS + MA* are further better than the results with only *CIS*. This shows that *CIS* is effective for multi-channel model training and *MA* can further improve the performance when it is used in the inference stage.

Table IV shows the recalls at breakeven points that are evaluated on an urban area of the test images. The area consists of patches where the number of background pixels is less than a third of the total number of pixels in a patch. In other words, the patches in the area always have both building and road pixels. In Table IV, all of the recall values of *ours (multi-channel with CIS + MA)* are better than for the single-channel model, with larger margins compared with the evaluations on the whole areas of the test images shown in Table III. This difference shows that our models have more advantage in urban regions that have both buildings and roads appearing at the same time within a small region because of utilization of the correlation. This advantage is important in some applications that require high accuracy in crowded urban areas, for example, real-estate management and updating of maps in areas that have many small houses and roads.

Focussing on the results in the road channel shown in Table IV, all scores are decreased from the results shown in Table III. The percentages of the decreases of *ours (single-channel with MA)*, *ours (multi-channel with MA)*, and *ours (multi-channel with CIS + MA)* are 5.53%, 3.53%, and 1.07%, respectively. The first two models show larger performance decreases compared with the last model, so that the extracted regions might have much difficulty in terms of extracting roads. Therefore, even in the case of the last model, the performance was decreased slightly. However, the last model has succeeded in suppressing the performance decrease compared with the first two models, with over five times and three times smaller percentage. This result also shows the effectiveness of CIS.

The additional computational cost of CIS has little influence on the training time and the processing time to output predicted label patches. In the inference stage, all of the processing times to predict a label image from a 1500 × 1500-sized aerial image were almost the same between all models when calculated on an NVIDIA Tesla K80. The average processing times of *ours (multi-channel with CIS)* and *Mnih-CNN* were 2.76 s and 2.80 s, respectively, for each aerial image. However, although the conventional method[10] needs two different models to create a multi-channel result, our model can predict building and road labels simultaneously with a single feedforward. This means that the conventional method needs about twice the processing time to obtain building and road predictions compared with ours. However, if we perform MA by using eight different versions of the model to obtain more accurate results, it requires about eight times longer processing time compared with the case of the conventional method. However, if multiple GPUs are available, all the calculations of the eight versions can be parallelized, so the drawback turns out to be marginal.

CONCLUSION

In this article, we proposed a method to train CNNs for multi-label semantic segmentation of aerial imagery, and a new output function *channel-wise inhibited softmax (CIS)* to train the CNNs effectively in such task. Furthermore, we showed that the new model averaging technique called *model averaging with spatial displacement* can improve the prediction results further. Our method with all proposals can predict building and road labels simultaneously with high accuracy compared with the conventional method[10] which

DOI: 10.2352/ISSN.2470-1173.2016.10.ROBVIS-392

predicts each label separately. We showed the performance of our methods by evaluating them on a new dataset *Mass. BR*. Furthermore, to show the benefit of using the correlation between buildings and roads in aerial images, we evaluated them on urban regions of test images and showed the effectiveness of the multi-channel formulation with CIS. In the case of road prediction, the results also showed that CIS enables us to achieve better results with a one-eighth sized dataset compared with the previous work.

Finally, we implemented our models with Caffe,[29] a deep learning framework, and all of the codes of our proposed methods, the experiments, and the new dataset *Mass. BR* are publicly available at https://github.com/mitmul/ssai.

ACKNOWLEDGMENTS

This work was partially supported by JST CREST "Intelligent Information Processing Systems Creating Co-Experience Knowledge and Wisdom with Human–Machine Harmonious Collaboration."

REFERENCES

[1] B. Sirmacek and C. Unsalan, "A probabilistic framework to detect buildings in aerial and satellite images," IEEE Trans. Geosci. Remote Sens. **49**, 211–221 (2011).
[2] C. Harris and M. Stephens, "A combined corner and edge detector," *Alvey Vision Conf.* (Manchester, UK, 1988), Vol. 15, p. 50.
[3] C. Unsalan, "Gradient-magnitude-based support regions in structural land use classification," IEEE Geosci. Remote Sens. Lett. **3**, 546–550 (2006).
[4] M. Vetterli and J. Kovačević, *Wavelets and Subband Coding* (Prentice-Hall, Englewood Cliffs, NJ, 1995), Vol. 87.
[5] E. Rosten, R. Porter, and T. Drummond, "Faster and better: a machine learning approach to corner detection," IEEE Trans. Pattern Anal. Mach. Intell. **32**, 105–119 (2010).
[6] C. Senaras, M. Ozay, and F. T. Yarman Vural, "Building detection with decision fusion," IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens. **6**, 1295–1304 (2013).
[7] C. J. Tucker, "Red and photographic infrared linear combinations for monitoring vegetation," Remote Sens. Environ. **8**, 127–150 (1979).
[8] N. Otsu, "A threshold selection method from gray-level histograms," Automatica **11**, 23–27 (1975).
[9] M. Ozay and F. T. Yarman Vural, "A new fuzzy stacked generalization technique and analysis of its performance" (2012), arXiv:1204.0171.
[10] V. Mnih, "*Machine Learning for Aerial Image Labeling*," Ph.D. thesis (University of Toronto, 2013).
[11] V. Mnih and G. Hinton, "Learning to detect roads in high-resolution aerial images," *Proc. 11th European Conf. on Computer Vision (ECCV)* (2010).
[12] V. Mnih and G. Hinton, "Learning to label aerial images from noisy data," *Proc. 29th Annual Int'l Conf. on Machine Learning (ICML 2012)* (2012).
[13] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," Science **313**, 504–507 (2006).
[14] S. Saito and Y. Aoki, "Building and road detection from large aerial imagery," Proc. SPIE **94050K**, (2015).
[15] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in Neural Information Processing Systems* (2012), pp. 1097–1105.
[16] A. Toshev and C. Szegedy, "Deeppose: human pose estimation via deep neural networks," *2014 IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)* (IEEE, Piscataway, NJ, 2014), pp. 1653–1660.
[17] C. Farabet, C. Couprie, L. Najman, and Y. LeCun, "Learning hierarchical features for scene labeling," IEEE Trans. Pattern Anal. Mach. Intell. **35**, 1915–1929 (2013).
[18] K. Fukushima, "Neocognitron: a self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position," Biol. Cybern. **36**, 193–202 (1980).
[19] D. H. Hubel and T. N. Wiesel, "Receptive fields, binocular interaction and functional architecture in the cat's visual cortex," J. Physiol. **160**, 106 (1962).
[20] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition," Neural Comput. **1**, 541–551 (1989).
[21] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," Proc. IEEE **86**, 2278–2324 (1998).
[22] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," Nature **323**, 533–536 (1988).
[23] L. Bottou, "Stochastic gradient learning in neural networks," Proc. Neuro-Nımes **91**, (1991).
[24] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors" (2012). arXiv:1207.0580.
[25] I. J. Goodfellow, D. Warde-Farley, M. Mirza, A. Courville, and Y. Bengio, "Maxout networks," *Proc. 30th Int'l. Conf. on Machine Learning (ICML)* (2013), pp. 1319–1327.
[26] C. Wiedemann, C. Heipke, H. Mayer, and O. Jamet, "Empirical evaluation of automatically extracted road axes," *Empirical Evaluation Techniques in Computer Vision* (1998), pp. 172–187.
[27] B. D. Ripley, *Pattern Recognition and Neural Networks* (Cambridge University Press, 1996), p. 150.
[28] L. Nonboe Andersen, J. Larsen, L. Kai Hansen, and M. Hintz-Madsen, "Adaptive regularization of neural classifiers," *DTIC Document* (1997), pp. 24–33.
[29] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: convolutional architecture for fast feature embedding" (2014), arXiv:1408.5093.