ISPRS Journal of Photogrammetry and Remote Sensing 130 (2017) 139–149

Contents lists available at ScienceDirect

ISPRS Journal of Photogrammetry and Remote Sensing

journal homepage: www.elsevier.com/locate/isprsjprs

# Simultaneous extraction of roads and buildings in remote sensing imagery with convolutional neural networks

Rasha Alshehhi [a,*], Prashanth Reddy Marpu [a], Wei Lee Woon [a], Mauro Dalla Mura [b]

[a] Institute Center for Smart and Sustainable Systems, Masdar Institute of Science and Technology, Abu Dhabi, United Arab Emirates
[b] GIPSA-lab, Grenoble Institute of Technology, Grenoble, France

ARTICLE INFO

*Article history:*
Received 3 January 2017
Received in revised form 29 April 2017
Accepted 2 May 2017
Available online 9 June 2017

*Keywords:*
Convolutional neural network
Low-level features
Adjacent regions
Extraction

ABSTRACT

Extraction of man-made objects (e.g., roads and buildings) from remotely sensed imagery plays an important role in many urban applications (e.g., urban land use and land cover assessment, updating geographical databases, change detection, etc). This task is normally difficult due to complex data in the form of heterogeneous appearance with large intra-class and lower inter-class variations. In this work, we propose a single patch-based Convolutional Neural Network (CNN) architecture for extraction of roads and buildings from high-resolution remote sensing data. Low-level features of roads and buildings (e.g., asymmetry and compactness) of adjacent regions are integrated with Convolutional Neural Network (CNN) features during the post-processing stage to improve the performance. Experiments are conducted on two challenging datasets of high-resolution images to demonstrate the performance of the proposed network architecture and the results are compared with other patch-based network architectures. The results demonstrate the validity and superior performance of the proposed network architecture for extracting roads and buildings in urban areas.


## 1. Introduction

Automated extraction of multiple urban objects from aerial and satellite imagery is an important step in many applications such as infrastructure planning, updating geographical databases, land use analysis and change detection. Classifying pixels into semantic objects in aerial and satellite images of urban areas is one of the most challenging and important problems. This is because the remote sensing images are usually characterized by complex data in the form of heterogeneous regions with large intra-class variations and often lower inter-class variations. This is even more prominent with urban objects such as buildings and roads. Such heterogeneity in remote sensing images restricts most of the existing methods that often depend on a set of predefined features, which in turn are extracted using tunable parameters. As a result, it is highly difficult to design a method which could achieve high accuracy, especially with increasing spatial resolutions. Recently, deep Convolutional Neural Network (CNN) architectures (Krizhevsky et al., 2012; Lin et al., 2014; Zeiler and Fergus, 2014; Szegedy et al., 2015; Liu and Deng, 2015; Zhou et al., 2014) have been achieving impressive state-of-the art performance for semantic classification, not only in remote sensing applications (Firat et al., 2014; Makantasis et al., 2015; Paisitkriangkrai et al., 2015; Kampffmeyer Michael and Jenssen, 2015; Jiang et al., 2015; Vakalopoulou et al., 2015; Castelluccio et al., 2015; Sherrah, 2016; Nogueira et al., 2016), but also in some networks that have been proposed in computer vision, e.g., fully connected network (Long et al., 2015), SegNet (Badrinarayanan et al., 2015), ReSeg (Visin et al., 2015), DeepLab (Chen et al., 2015; Papandreou et al., 2015), Deconvolutional network (Noh et al., 2015), Decoupled Network (Hong et al., 2015), Patch Network (Brust et al., 2015; Brust et al., 2015), Deep Parsing Network (Liu et al., 2015), integrated CNN with CRFs (Chen et al., 2015; Zheng et al., 2015; Lin et al., 2016) and combined CNN with segmentation (Zhao et al., 2015; Kim et al., 2015).

Deep CNN architecture is quickly becoming prominent in remote sensing applications since it has the ability to effectively encode spectral and spatial information based on the input image data, without any prepossessing step. It consists of multiple interconnected layers and learns a hierarchical feature representation from raw pixel-data. It discovers features in multiple levels of

* Corresponding author.
*E-mail addresses:* ralshehhi@masdar.ac.ae (R. Alshehhi), pmarpu@masdar.ac.ae (P.R. Marpu), wwoon@masdar.ac.ae (W.L. Woon), mauro.dalla-mura@gipsa-lab.grenoble-inp.fr (M.D. Mura).

http://dx.doi.org/10.1016/j.isprsjprs.2017.05.002

representations. The lowest level is depicted by the primitive features of pixels (e.g., spectral properties) and the higher level involves transforming from raw pixel representation into gradually more abstract representations that are invariant to small geometric variations (e.g., edges and corners), and further transforming them gradually to make them invariant to contrast changes and contrast inversion (e.g., object parts). At the end, the most frequent patterns related to more abstract categories associated with whole objects are identified.

There have been several methods of CNN architectures in remote sensing. Paisitkriangkrai et al. Paisitkriangkrai et al., 2015 combined simple features (e.g., Digital Surface Model (DSM) and Normalized DSM) with multi-resolution CNN features to detect multiple classes using multiple binary classifiers. They applied multi-class concatenation classifier on CNN features and then applied pixel-based Conditional Random Field (CRF) classifier as the post-processing stage to smoothen the final pixel-based classification. In Sherrah, 2016, all convolution layers in CNNs are replaced with fully connected layers, and down-sampling pooling is replaced with no down-sampling pooling. Kampffmeyer Michael and Jenssen (2015) applied similar deep CNN architecture to extract small objects (e.g., cars), which have lower class distribution by combination with deconvolution layers. In Jiang et al. (2015), graph-based segmentation (Felzenszwalb and Huttenlocher, 2004) is integrated with CNNs to localize image patches,[1] which help in localizing vehicles effectively. In Lngkvist et al. (2016), CNNs are integrated with spectral features of Simple Linear Iterative Clustering (SLIC) segmentation (Achanta et al., 2012) in the post processing stage to improve the performance of CNNs.

In general, CNN architectures for semantic pixel-based classification use two main approaches: patch-based and pixel to pixel based (end to end). Patch-based methods commonly start with training of CNN classifier on small image patches and then predict the class of each pixel, using a sliding window approach. Alternatively, the fully connected layers can be converted to convolution layers, avoiding overlapping computations required for each pixel (Paisitkriangkrai et al., 2015; Sherrah, 2016). This approach is usually used to detect large urban objects. Pixel-based methods use an end to end CNN, where usually Fully Convolutional Network (FCN) or encoder-decoder architectures are used by applying upsampling, interpolation, etc. (Jiang et al., 2015; Lngkvist et al., 2016). This approach is important to detect fine detail of the input images.

In this work, we propose a modified patch-based CNN architecture to simultaneously extract roads and buildings from satellite imagery by replacing fully connected layers with Global Average Pooling (GAP) (Lin et al., 2014; Szegedy et al., 2015; Zhou et al., 2015), which considers an average of all feature maps from the last convolution layer of the CNN. We concentrate on roads and buildings because these classes make up a large portion of urban fabric. Moreover, they exhibit a significant amount of urban structure that can be exploited to improve classification in noisy data. As a post-processing step, Simple Linear Iterative Clustering (SLIC) segmentation (Achanta et al., 2012) is applied on the CNN probability map. The shape features of adjacent SLIC regions of roads and buildings are used to link discontinuous road segments and to merge misclassified regions of buildings.

The remainder of this paper is organized as follows. Section 2 introduces some of the related works that used CNNs in extracting roads and buildings and other works which are more related to the proposed CNN architecture. An overview of the proposed CNN is presented in Section 3. Section 4 presents the experimental results and Section 5 summarizes the most important findings.

[1] Image windows with predefined dimensions.

## 2. Related works and contribution

In this section, some promising CNN approaches for extracting roads and buildings from aerial imagery are discussed, highlighting their main contributions. Some related CNN architectures in computer vision applications are also summarized and finally contributions of this paper are outlined.

There is a significant amount of literature on semantic pixel-based classification for extraction of roads and buildings in remote sensing imagery. Mnih (2013) proposed a road extraction method based on patch-based CNN. The CNN input is extracted from Principal Component Analysis (PCA) features. Then the PCA vectors are trained by the Restricted Boltzmann Machine (RBM) and refined by a post-processing network to incorporate structure such as road connectivity into the final road network. Shu (2014) illustrated the main differences between the performance of the CNN architecture and image segmentation on the same dataset. Saito and Aoki (2015), Saito et al. (2016) used a single CNN architecture for extracting roads and buildings on the Mnih imagery dataset (Mnih, 2013), where each image consists of RGB channels. The CNN predicts a multi-class probability output of roads, buildings and background simultaneously. They also applied Channel-wise Inhibited Softmax (CIS) function to suppress the effect of the background.

Maggiori et al. (0000) suggested a similar architecture as (Shu, 2014) to detect buildings. However, they used a pixel-based approach by applying deconvolution operators, which uses upsampling into the initial resolution to produce dense pixel-based classification. To detect buildings, Marcu and Leordeanu (0000) proposed a dual-stream deep network model to extract roads and buildings separately based on Alex-Net (Liu and Deng, 2015)[2] and VGG-Net (Liu and Deng, 2015).[3] Alex-Net considers information from large areas around the object of interest due to the larger filter size. VGG-Net network focuses on local and object level information due to the smaller filter size. Both networks are combined into final subnet, composed of three Fully Connected (FC) layers.

In computer vision, CNN architectures such as Network in Network (NIN) (Lin et al., 2014) and GoogLetNet (Szegedy et al., 2015) proposed avoiding the use of Fully connected layers to minimize the number of parameters while maintaining the high performance. In Lin et al. (2014) and Szegedy et al. (2015), global average pooling is used to act as a structural regularizer, preventing overfitting during training. Zhou et al. (2015) revisited (Lin et al., 2014) and showed that convolution units have the ability to localize objects in convolution layers; however, this ability is lost when fully connected layers are used. Therefore, they use global average pooling and are able to achieve lower error for object localization. Another similar approach is based on global maximum pooling by Oquab et al. (2015). They applied global maximum pooling to localize a point lying on object boundaries, rather than the complete extent of the objects.

In this work, we follow the same approach as (Lin et al., 2014; Szegedy et al., 2015; Zhou et al., 2015; Oquab et al., 2015) for extracting roads and buildings from two challenging datasets with different spatial image resolutions and different conditions. This work proposes a multi-class prediction method with a single CNN architecture by predicting three different classes simultaneously

[2] Alex-Net, proposed by Krizhevsky et al. (2012), was the winner of ImageNet Large Scale Visual Recognition Challenge (ILSVRC) Deng et al., 2009. It consists of five convolution layers, some of which are followed by max-pooling layers, and three fully connected layers with a final softmax.

[3] VGG-Net, presented in (Liu and Deng, 2015), won the localization and classification tracks of the ILSVRC-2014 competition. It has thirteen convolution layers, five pooling ones and three fully connected one with a final softmax.

in various datasets with different image resolutions. The proposed CNN architecture uses a deeper patch-based segmentation, and simple global average pooling layer is used instead of fully connected layers. It introduces a new post-processing method based on low-level features (e.g., asymmetry and compactness) of adjacent regions with the lower probability to be classified as roads or buildings by incorporating spatial structure (e.g., road connectivity and building closure). Roads are considered to consist of long and thin segments which should form connected objects and buildings are considered as relatively compact regions.

## 3. Proposed method

In this section, the proposed framework for extracting roads and buildings in the high-resolution imagery of urban areas is illustrated. The method does not require any pre-processing stage. First, Convolutional Neural Network (CNN) architecture and learning framework are discussed. Second, some spatial features of adjacent SLIC regions, which are used to enhance CNN outputs, are presented.

### 3.1. Convolutional Neural Network (CNN) architecture

The CNN architecture often consists of alternatively stacked convolution layers followed by fully connected layers, as illustrated in Fig. 1.

The convolution layer usually consists of different operators: convolution, non-linear transformation and pooling. The convolution produces new images (called feature maps), each element of which is obtained by computing a dot product between the local region (receptive field) it is connected to in the input feature maps and a set of weights (called filters or kernels). This operator is followed by a elementwise non-linear function (e.g. ReLU, tanh, etc.) and then by a pooling function. The pooling performs a sampling along the spatial dimensions of feature maps via predefined function (e.g. maximum, average, etc.) on a local region (Hu et al., 2015; Nogueira et al., 2016).

The fully connected layer takes all neurons of the previous layer, either the convolution layer or another fully connected layer, and connects them to every single neuron in its layer. In order to reduce overfitting in fully connected layers, dropout regularization method (Srivastava et al., 2014) is usually employed. It randomly drops several neuron outputs or decreases the number of neurons of the network that do not contribute to the forward-pass and back-propagation anymore (Krizhevsky et al., 2012; Lin et al., 2014; Zeiler and Fergus, 2014; Szegedy et al., 2015; Liu and Deng, 2015; Zhou et al., 2014; Nogueira et al., 2016; Mnih, 2013; Shu, 2014). how to define "not contribute"

The output of the fully connected layer is used to produce probabilistic output for each class (classifier layer). The most common transformation function for multi-class classification is the softmax function (Bengio, 2009). It is a multinominal logistic function which generates a vector in the range (0,1) representing a categorical probability distribution for every class (Krizhevsky et al., 2012; Lin et al., 2014; Zeiler and Fergus, 2014; Szegedy et al., 2015; Liu and Deng, 2015; Zhou et al., 2014; Nogueira et al., 2016; Mnih, 2013; Shu, 2014).

A convolution layer takes a $W \times H$ image patch with $N$-channels centered at $x(i,j)$ and two-dimensional filter-kernel $w_f \times h_f$ as inputs and outputs feature maps of $(W - w_f + 1) \times (H - h_f + 1)$ with $K$-channels. Each channel of this output image is called a filter site. The output of the convolution process is effected by a stride $s_f$ parameter. Stride $s_f$ is the distance that is required to slide convolution process in the input image or feature map (Nogueira et al., 2016). If $s_f > 1$, the size of an output map from convolution process is decreased to $((W - w_f)/s_f + 1) \times ((H - h_f)/s_f + 1)$. The convolution process is defined as follows:

$$x_k(ii,jj) = \sum_{n=1}^{N} \left\{ \sum_{p=0}^{w_f-1} \sum_{q=0}^{h_f-1} x_n(i \cdot s_f + p, j \cdot s_f + q) \cdot h_k(p,q) \right\} + b_k, \qquad (1)$$

where $x_n(i,j)$, $x_k(ii,jj)$, $h_k(p,q)$ and $b_k$ are pixel value at $(i,j)$ in $n$-th channel of an input image or of a feature map, pixel value at $(ii,jj)$ of a feature map on $k$-th filter site, a weight value at $(p,q)$ on $k$-th filter, and a bias parameter of $k$-th filter that is shared among all locations $(p,q)$, respectively. Fig. 2 illustrates the main concept of the convolution operator.

A convolution operator is followed by transformation, called activation function. Let us assume $x_k(ii,jj)$ as an input to the activation function of neural network, which is output of convolution process. $w$ is a weight vector and $b$ is a bias vector. The activation function is expressed as:

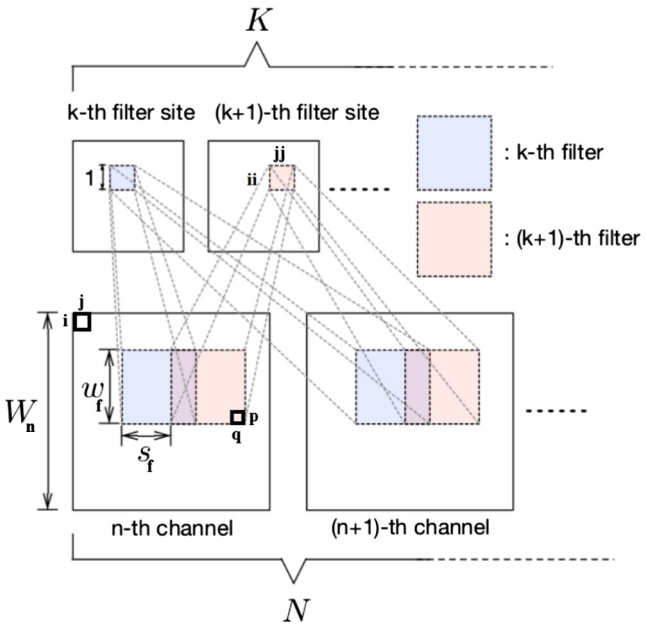



**Fig. 2.** An overview of the convolution operator (modified from Saito and Aoki (2015)).

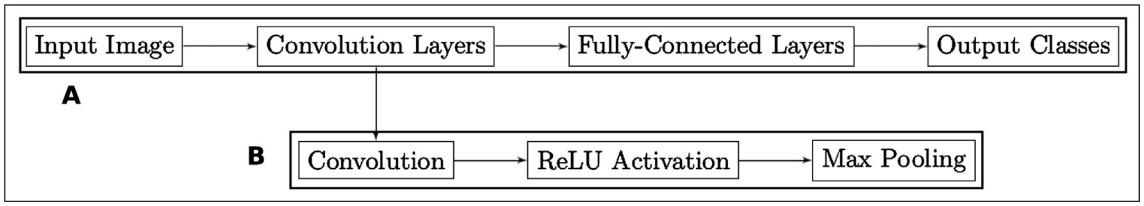



**Fig. 1.** The standard architecture of the CNN. (A) Presents the main components of CNN architecture, and (B) presents the main components of the convolution layer.

$$Z(x_k(ii,jj)) = f\left(\sum_{k=1}^{K} x_k(ii,jj)\cdot w_k + b_k\right) \Longleftrightarrow Z = f(X\cdot W + b), \quad (2)$$

There are a lot of alternative functions for $f(\cdot)$ such as tanh, sigmoid, hyperbolic and rectified functions. The rectified function is currently the most used in the literature because neurons, with rectified function, work better to avoid saturation during the learning process, induce the sparsity in the hidden units and do not face gradient vanishing problem, which occurs when the gradient norm becomes smaller after successive updates in the back-propogation process. In this paper, we use Rectified Linear Unit (ReLU) function (Nair and Hinton, 2010), as follows:

$$A(x_k(ii,jj)) = \max(0, Z(x_k(ii,jj))), \quad (3)$$

A pooling operator performs spatial subsampling by considering maximum or average value of $w_p \times h_p$ pooling window. This operator ensures that same result can be obtained, even when image features have small translation or rotation. Let us assume that $A(x_k(ii,jj))$ is an output of the previous activation operator and by applying max-pooling with a stride interval $s_p$, the output $x_k(ii_p, jj_p)$ is expressed as follows:

$$x_k(ii_p, jj_p) = \max_{0\leqslant ii_p\leqslant h_p-1, 0\leqslant jj_p\leqslant w_p-1} A(x_k(ii,jj)), \quad (4)$$

In max-pooling process, the input of $k$-th channel with size $((W-w_f)/s_f+1)\times((W-w_f)/s_f+1)$ is downscaled to the size of $((W-w_f)/s_f\cdot s_p+1)\times((W-w_f)/s_f\cdot s_p+1)$. Fig. 3 shows the main idea of the max-pooling operator.

After convolution and fully connected layers, a classifier layer is used to predict class probabilities, which is represented as multi-channel patch $\hat{m}$ of the input image patch $n$ and ground-truth image patch $\tilde{m}$. The most common transformation function for multi-class predication is the softmax function. Let us assume $W_m \times H_m \times K$ is the form of reshaped output of the CNN, where $K$ is the number of channels of the output image patch $\hat{m}$. $x = [x_1, \ldots, x_K]^T$ denotes pixel value in the output of fully connected layer and softmax function is applied to each $x$ to convert into probability vector $\hat{m} = [\hat{m}_1, \ldots, \hat{m}_K]^T$ as follows:

$$\hat{m}_{w,b} = \frac{\exp(x\cdot w_c)}{\sum_k^K \exp(x\cdot w_k)}, \quad (5)$$

Fig. 4 shows the proposed CNN architecture. In this work, the architecture consists of five convolution layers, where the first and second layers are separated by a max-pooling layer. Five convolution layers are followed by an average pooling layer followed by softmax function.

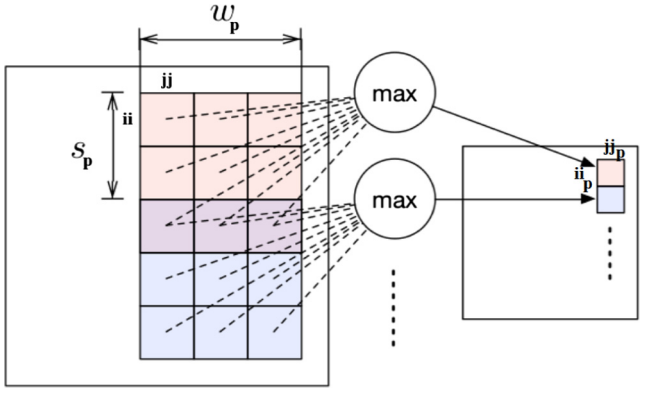



**Fig. 3.** An overview of the max-pooling operator (modified from Saito and Aoki (2015)).

why a vector can be shaped to get three channels for road, bldg and background?

In a similar way to what has been proposed by Mnih (2013), Shu (2014) and Saito and Aoki (2015, 2016), the input image is divided into $64\times64$ patches, each with three channels (Red Green and Blue), and the output is a 768-dimensional vector, which is reshaped into $16\times16$ of three channels (Road, Building and Background). The decision of selecting input patch wider than output patch is that larger contextual information could be utilized to predict the final probability map and consequently it is easier to recognize bigger urban classes (e.g. buildings) (Mnih, 2013; Saito et al., 2016). The decision on selecting simultaneous multi-class prediction strategy is that it will be more accurate compared to predicting a single-class independently, if the correlation is exploited effectively in CNNs (Saito and Aoki, 2015; Saito et al., 2016). Each input patch is normalized by subtracting the mean value and divided by the standard deviation (global contrast normalization) (Mnih, 2013; Long et al., 2015; Saito and Aoki, 2015; Saito et al., 2016).

The last convolution layer is followed by Global Average Pooling (GAP). GAP simply averages the feature maps where similar results are expected in a patch. However, traditional fully connected layer is a result of mapping all feature maps of the last convolution layer to one layer (Lin et al., 2014). Fully connected layer causes overfitting because of parameters. To reduce overfitting, it heavily depends on dropout regularization which randomly sets half or more of the activations of the fully connected layers to zero during training and it requires optimization in tuning the parameters. However, GAP does not require any optimization (Lin et al., 2014; Zhou et al., 2015; Zhou et al., 2015).

As demonstrated in Fig. 4, an input patch is 3@64 × 64, [4] consisting of three channels, each with dimension $64\times64$. The first convolution layer is 128@14 × 14, composed of 128 filter-channels, each with a dimension of $14\times14$, which is a result of the convolution of an input patch with the kernel of dimension $12\times12$ with stride 4, followed by max-pooling of dimension $2\times2$ resulting in 128@13 × 13. This process is followed by convolution of 128@13 × 13 with filter-kernel of sizes $5\times5$, $3\times3$, $3\times3$ and $3\times3$ with filter units 256, 512, 32 and 768, yielding outputs of sizes 256@9 × 9, 512@7 × 7, 32@5 × 5 and 768@3 × 3, respectively. All convolution layers have a stride of 1, except the first one, which has a stride of 4. GAP is computed over all 768 channels of dimension $3\times3$ to reproduce 768@1 × 1, which is then reshaped into 3@16 × 16.

The CNN is trained by minimizing the negative log likelihood using mini-batch stochastic gradient descent with momentum (Bengio, 2012) and its loss function is defined with the summation of pixel-wise cross entropy between predicted label-probability and true ground-truth patches.

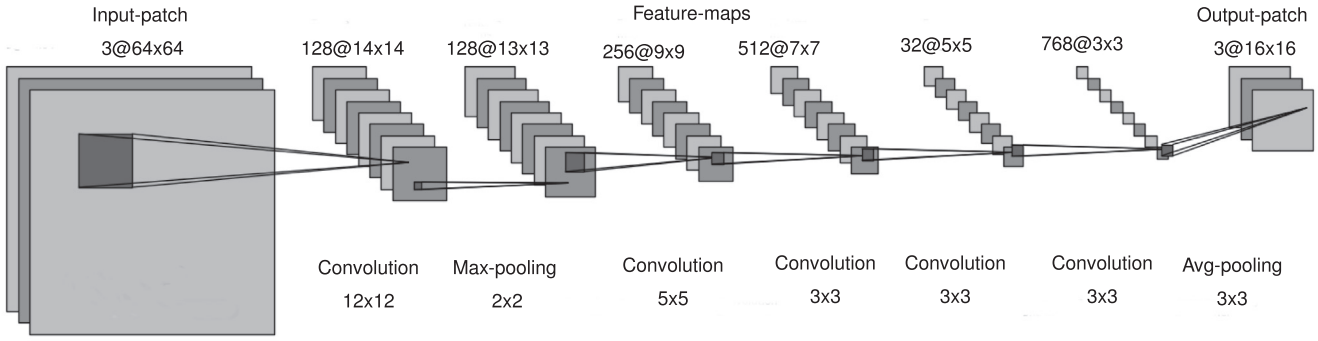
the number of filter units is determined by authors

### 3.2. Post-processing

The convolutional network detects road regions, but it does not guarantee continuous road regions, especially around road intersections. Similarly, with building regions, it does not guarantee compacted contours and hence may produce irregular building outlines. Therefore, a post-processing step is used to reduce misclassified regions to better represent real-world objects.

In this work, Simple Linear Iterative Clustering (SLIC) (Achanta et al., 2012) is applied to obtain the initial segmented image based on similarity in [lab] color space and proximity in the [xy] image plane. To introduce adjacent relationships between previous segments, Region Adjacent Graph (RAG) (Tremeau and Colantoni,

[4] @: this symbol is used to differentiate between the number of channels and dimension of the input image or between the number of filter units and feature maps.




128 is produced by 128 kernel filters, each kernel filter will convolute each input channel, and combine the 3 outputs into one layer of the 128 layers, so the number of the channels of the output is determined by the number of kernel filters, not the number of channels of the input image

number of filter units is in fact the number of kernel filters

**Fig. 4.** The architecture of the used CNN.

2000) is used to facilitate merging process between superpixels as follows:

1. If adjacent regions are classified as the same class with the high CNN certainty ($P > T_{max}$) with similar brightness, then those segments probably belong to the same real-world objects and hence should be merged.
2. Roads are narrow elongated regions (Liu et al., 2015; Jabari and Zhang, 2013); therefore, road superpixels, with low CNN certainty ($T_{min} \leqslant P \leqslant T_{max}$) and adjacent to road regions of high certainly in road-output channel with identical asymmetry and elongation, are classified as roads.

$$(EI \geqslant T_{EI}) \cap (AI \geqslant T_{AI}), \tag{6}$$

$$EI = \frac{I_{major}}{I_{minor}}, \tag{7}$$

$$AI = \frac{2\sqrt{\frac{1}{4}(\sigma_X^2 + \sigma_Y^2)^2 + (\sigma_{XY}^2)^2 - \sigma_X^2\sigma_Y^2}}{\sigma_X^2 + \sigma_Y^2}, \tag{8}$$

where $EI$ is Elongation Index, $AI$ is Asymmetry Index, $T_{EI}$ and $T_{AI}$ are corresponding thresholds, which determined by the mean of road regions with high probabilities. $I_{major}$ is the length of the major axis of a region and $I_{minor}$ is the length of the minor axis of a region. $\sigma_X^2$ is variance of $X$ of a region and $\sigma_Y^2$ is variance of $Y$ of a region (Liu et al., 2015; Yu et al., 2016), where $(X, Y)$ corresponds to the pixel-coordinate.

3. Buildings are compact blobs with high density (Liu et al., 2015; Jabari and Zhang, 2013); therefore, a building superpixels, with low certainty ($T_{min} \leqslant P \leqslant T_{max}$) and nearby by building regions with identical compactness and density features, are classified as parts of buildings.

$$(CI \geqslant T_{CI}) \cap (DI \geqslant T_{DI}), \tag{9}$$

$$CI = \frac{2\sqrt{A\pi}}{P}, \tag{10}$$

$$DI = \frac{\sqrt{N}}{1 + \sqrt{\sigma_x^2 + \sigma_y^2}}, \tag{11}$$

where $CI$ is Compactness Index, $DI$ is Density Index, $T_{CI}$ and $T_{CI}$ are corresponding thresholds, which are determined by the mean of building regions. $A$ is the area of a region and $P$ is a perimeter of a region. $N$ is number of pixels in a region and $\sigma_x$ and $\sigma_y$ are standard deviation of $X$ and $Y$ (Liu et al., 2015; Yu et al., 2016). $T_{max}$ and $T_{min}$ are defined as probability values which are higher than mean, and between mean and mean-half, respectively.

## 4. Experimental analysis and discussion

To verify the effectiveness of the proposed method, extensive experiments to extract roads and buildings from remote sensing images have been conducted on two datasets. The proposed CNN architecture is compared with the same architecture with alternative sub-components and also with other CNN architectures. In this section, the experimental setup is described and experimental results are illustrated.

### 4.1. Experimental setup

- Datasets description
  1. Massachusetts dataset
     Massachusetts dataset consists of 137 training, 4 validation and 10 testing images (Table 1). The size of each image is $1500 \times 1500$ pixel with the spatial resolution of 1 meter per pixel, composed of red, green, blue channels. This dataset was built by Mnih (2013). The ground-truth of the images consists of three classes i.e., roads, buildings and background, and it was produced by Saito et al. (2016).
  2. Abu Dhabi dataset
     The multi-class ground-truth dataset, composed of roads, buildings and background, is built to match images acquired over Abu Dhabi. The extent of the dataset of dimension $41,411 \times 31,894$ pixels is divided into three sets: training (150 images), validation (30 images) and testing (30 images) with no overlapping areas (Table 1). The size of all images in this dataset is $1500 \times 1500$ composed of red, green and blue channels and the spatial resolution is 0.5 meter per pixel.
- Experimental setting
  All experimental parameters for CNNs are chosen after extensive experiments with various values and selecting the ones with the highest performance. Training is carried out by optimizing the logistic regression function using stochastic gradient descent (Bengio, 2012) and mini-batch size of 128 with the momentum of 0.9. The training was regularized by weight decay set to 0.0005, and dropout regularization for all fully connected layers with dropout ratio set to 0.5. We initialized the weights in each layer with a random number drawn from a zero-mean Gaussian distribution with standard deviation 0.01. The learning rate is started with 0.0005 with initial bias set to constant 0.1. Fig. 5 show the loss function in each of training and validation dataset in Abu Dhabi data after 400 epochs [5]. It is obvious that error is gradually decreased.
  In all experiments, the input patch is $3@64 \times 64$ and output patch is $3@16 \times 16$. All experiments in this paper were performed using

[5] epoch: each time we run the entire dataset.

**Table 1**
An overview of the datasets.

| Dataset | Training | Validation | Testing |
|---|---|---|---|
| Massachusetts road & building | 137 | 4 | 10 |
| Abu Dhabi road & building | 150 | 30 | 30 |

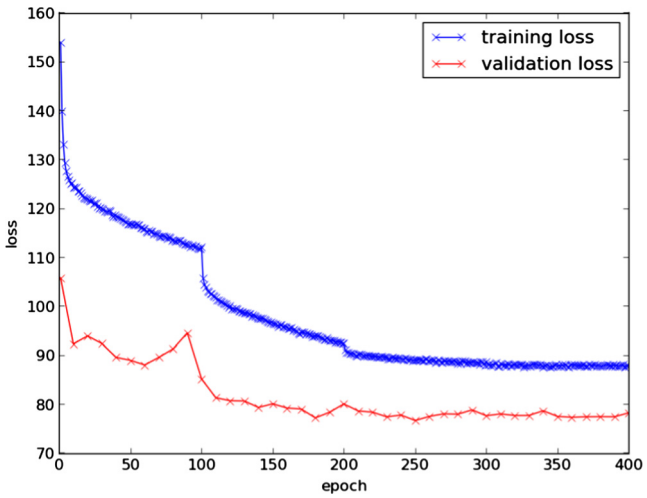



**Fig. 5.** Loss function in Abu Dhabi data.

the deep learning framework Caffe (Jia et al., 2014) and they are conducted on a NVIDIA 340.29, 2GPU, Tesla K20m of 4799MiB. The time to train the CNN model in 1 GPU is under five days. The maximum time requires to classify images is around 8.9 s.

The number of SLIC segments ($K$) is chosen according to the spatial resolution of the input image and to be reasonably high to facilitate over-segmentation. Road width is relatively small (e.g., $3 \leqslant w \leqslant 8$ m) in high-resolution images. If the spatial resolution of an image $1000 \times 1000$ is 0.5 m, where 3–8 m corresponding to 6–16 pixels, the number of superpixels can be roughly between ($K = \frac{1000\times1000}{6\times6} \approx 27000$) and ($K = \frac{1000\times1000}{16\times16} \approx 4000$). To be able to facilitate over-segmentation, $K$ is selected to be closer to the lower limit. Consequently, the size of each superpixel $s$ is computed based on the total number of pixels $N = 1000 \times 1000$ and the number of superpixels $K$ ($s = \frac{N}{K}$).

- Evaluation metrics

The most common metrics for evaluation for extracting roads and buildings are correctness (precision) and completeness (recall) (Mnih and Hinton, 2010,; Rajeswari et al., 2011,; Maurya et al., 2011,; Sujatha and Selvathi, 2015). Completeness is the fraction of true target-pixels that are within pixels of a predicted target-pixel; while, correctness measures the fraction of predicted target-pixels that are within pixels of a true target-pixel.

$$completeness = \frac{TP}{TP + FN}, \tag{12}$$

$$correctness = \frac{TP}{TP + FP}, \tag{13}$$

where $TP$ is defined as the number of target-pixels correctly detected, $FP$ is defined as the number of non-target pixels detected as target, $TN$ is the number of non-target pixels correctly detected, and $FN$ is the number of target-pixels detected as non-target pixels. In all experiments, the results at breakeven point are presented; i.e., where correctness and completeness are equal.

McNemar's statistical test (Leeuw et al., 2006) is also used to evaluate the performance of the proposed method. The McNemar's test is preferable because it is a parametric test, more precise and sensitive than Z-test, which is not appropriate if same samples are used in the comparison (Foody, 2004). The test is based on a chi-square statistic, computed from two error matrices using:

$$X^2 = \frac{|b - c| - 1}{\sqrt{b + c}}, \tag{14}$$

where $b$ and $c$ denote the number of cases correctly classified by method 2 but wrongly classified by method 1, and wrongly classified by method 2 but correctly classified by method 1. There are other cases which are $a$ and $d$, denoting the number of positive and negative cases that are correctly classified by method 1 and method 2.

- Compared methods

To verify the performance, the proposed CNN is compared with alternative CNN components and alternative post-processing stages, as follows:

1. Global Max-Pooling (GMP): it is identical to the proposed network, but GAP is replaced by GMP. This is because we believe that it is important to highlight the intuitive differences between GAP and GMP since GAP identifies the extent of objects; however, GMP identifies one discriminative part.
2. Fully Connected Layers (FCLs): it has similar network architecture; however, GAP is replaced with Fully Connected (FC) layers of 768 units. The fully connected layer is similar to GAP. It performs linear transformations of the vectorized feature maps. However, fully connected layers can have dense transformation matrices and values are subject to back-propagation optimization. The GAP is a prefixed transformation; where it is non-zero only on block diagonal elements which share the same value (Lin et al., 2014).
3. Multi-scales: the training on small patches is problematic because in high-resolution images such patches tend to cover fragmented buildings and thus fail to capture complete information about individual buildings. The output of the proposed network is combined with another network by considering the average of both networks. The input patch for another network is $3@128 \times 128$ and output patch is $3@16 \times 16$. In its first convolution layer, filter-kernel is $16 \times 16$, stride is 8 and max-pooling kernel is $3 \times 3$.
4. Conditional Random Field (CRF): the proposed network is followed by CRF model, proposed by (Krähenbühl and Koltun, 2011). The unary cost is based on the class probability from the proposed CNN classifier. The pairwise costs use a contrast-sensitive Potts model to penalize class boundaries with low contrast.

We also compare the proposed approach with other CNN architectures, as discussed in Section 2:

1. The method by Mnih (2013) is learned and evaluated on roads and buildings separately because they designed an architecture for a single-class prediction. The input patch is $3@64 \times 64$. The CNN architecture consists of three convolution layers ($64@13 \times 13$, $112@9 \times 9$ and $80@7 \times 7$) followed by two fully connected layers with 4096 units and 256 vector, which is reshaped to $1@16 \times 16$. In this paper, the results of integration of MLP convolution layers in the post-processing stage is illustrated, since it has the best performance.
2. The method by Shu (2014) is also for single-class prediction. The input patch is $3@64 \times 64$. The CNN architecture consists of three convolution layers ($64@9 \times 9$, $128@7 \times 7$ and $128@5 \times 5$) followed by two fully connected layers with 4096 units and 256 vector, which is reshaped to $1@16 \times 16$.

3. The method by Saito et al. (2016) is a multi-class prediction of roads, buildings and background. The input patch is 3@64 × 64. The CNN architecture consists of three convolution layers (64@13 × 13, 112@9 × 9 and 80@7 × 7) followed by two fully connected layers with 4096 units and 768 vector, which is reshaped to 3@16 × 16.

In order to compare our results with results reported by Mnih (2013),Shu (2014) and Saito et al. (2016), we use the same data, the same size of input patches (3@64 × 64), same output patches (1@16 × 16) and same metrics to evaluate our results.

### 4.2. Results

#### 4.2.1. Massachusetts data

Fig. 6 presents an example from Massachusetts data after applying convolutional network and low-level shape features. The CNN model achieves better performance after adding shape properties, as shown in Fig. 6.

The CNN model assigns same class to all buildings, and some adjacent small buildings appear as one connected region, which are difficult to separate from each other. Introduction of compactness and density features help to smoothen boundaries of large buildings. However, because of relatively large size of segments, adjacent small buildings are presented as one connected component. This may require another level of segmentation to differentiate buildings from the background (Matinfar et al., 2007; Jabari and Zhang, 2013; Yu et al., 2016). Additional post-processing helps to connects some disjointed road regions by considering pixels between adjacent regions with lower probability.

Table 2 presents a comparison between the proposed CNN with alternative CNN components in Massachusetts data according to correctness at the breakeven point. The results of GAP outperforms that of GMP. This is because, by finding the maximum of a map, low scores in all regions do not impact the final score except the most discriminative one. However, by finding the average of a map, the values are maximized by finding all discriminative parts of an object.

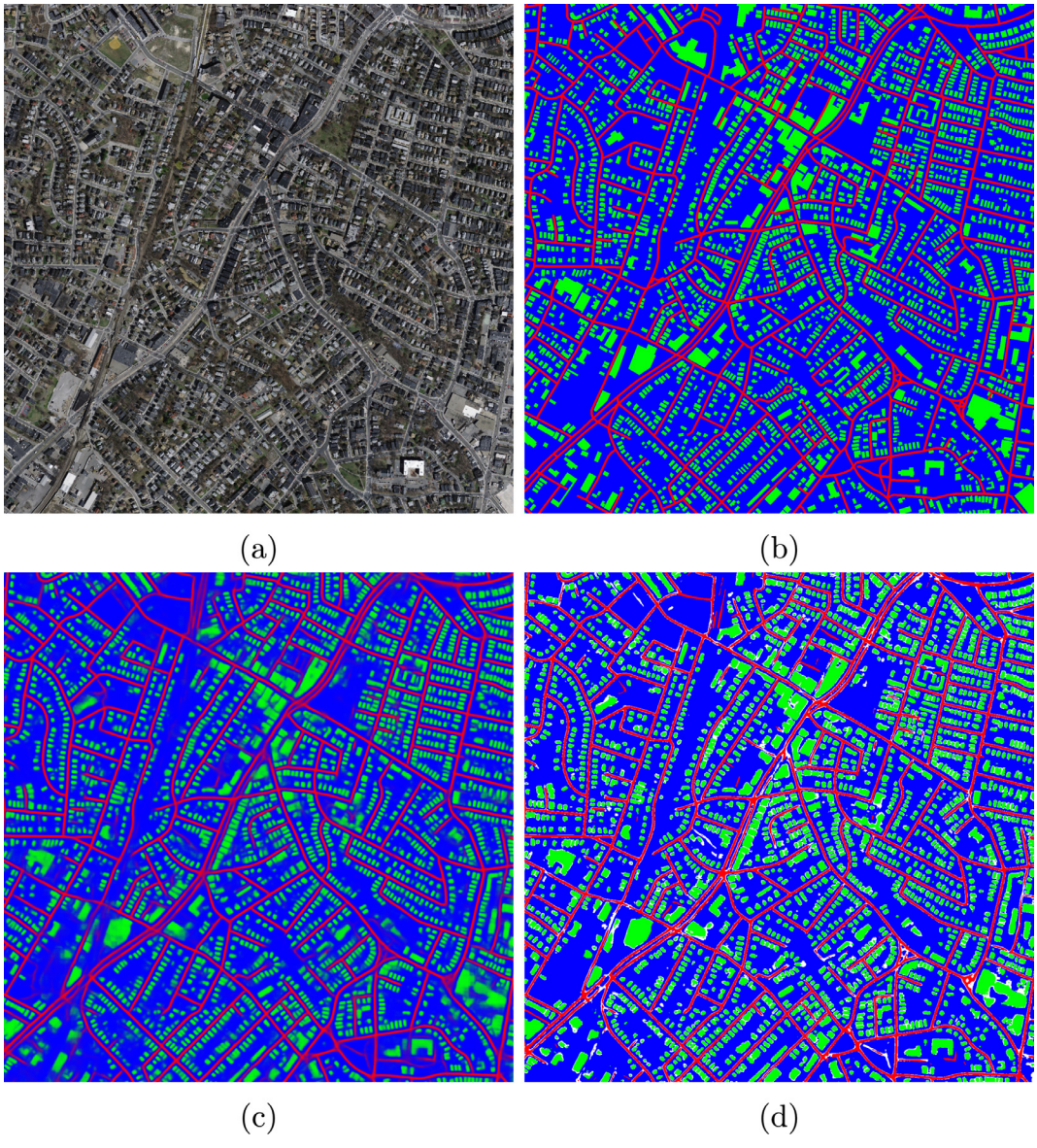



**Fig. 6.** An example from Massachusetts data. (A) Test image, (B) ground-truth image, (C) the result obtained by convolutional network, (C) the result obtained after post-processing stage: red: road with high probability, green: building with high probability, white: road/building with lower probability and considered as road/building. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

**Table 2**
Correctness (%) at breakeven of multi-class prediction in Massachusetts data.

| Dataset | Methods | Roads | Buildings |
|---|---|---|---|
| Massachusetts | Proposed method with GAP | 91.9 ± 2.1 | 94.8 ± 1.9 |
| | Proposed method with GMP | 88.7 ± 2.1 | 92.3 ± 2.0 |
| | Proposed method with FCLs | 90.7 ± 1.9 | 94.5 ± 2.2 |
| | Proposed method with GAP + Multi-scale | 92.3 ± 2.0 | 95.1 ± 2.1 |
| | Proposed method with GAP + CRF | 92.1 ± 2.1 | 95.0±2.5 |
| | Proposed method with GAP + segmentation | 93.1 ± 2.1 | 95.5 ± 2.0 |

**Table 3**
Correctness (%) at breakeven in the Massachusetts data by applying different methods.

| Dataset | Methods | Roads | Buildings |
|---|---|---|---|
| Massachusetts | Mnih (2013) | 90.1 | 92.0 |
| | Shu (2014) | 87.1 | 92.3 |
| | Saito et al. (2016) | 90.5 | 94.3 |
| | Proposed model | 91.7 | 94.6 |

In the case of replacing GAP with fully connected layers while other parts of the model remain the same, fully connected layers have shown higher testing error and consequently lower performance; However, adding dropout regularizer before fully connected layer improves results when compared with CNN without dropout. Moreover, when dropout probability is high (e.g., 0.7 and 0.8), the results fluctuate, because most of the features are dropped, and CNN cannot gather useful information to build a robust feature representation.

The improvement of multi-scale CNN over a single-scale CNN is because multi-scale CNNs smoothen the outputs over the boundaries of predicted label-probability patches. This is because it considers the average of discriminative parts of each urban-class.

Graph-based segmentation and Conditional Random Field (CRF) are used as post-processing methods to further improve the results. By considering dependencies between nearby pixels across patch with CRF model, multi-class prediction is improved. However, considering shape features of roads and buildings significantly improves the performance of CNN, especially at boundaries of buildings and road intersections.

Table 3 shows the performance of the single prediction (either roads or buildings) of the proposed method after the post-processing step, Mnih (2013), Shu (2014) and Saito et al. (2016) approaches in Massachusetts data according to correctness at breakeven point. The main difference between the proposed CNN and other approaches is the difference in the size of kernel filters, number of filter units, number of convolution layers and post-processing stage. Due to the fact that the GAP is a kind of classifier with no parameters to optimize, our method achieves the best performance. By replacing GAP with fully connected layers, as shown in previous Table 2 (Proposed model + FCLs) and comparing with Saito et al. (2016) method, it is observed that our network configuration has better performance, although the main difference is the number of convolution layers and number of filter units. This proves the theory, reported in Liu and Deng (2015), that deeper network configuration (additional convolution layers, filter units, etc.) is better as it yields more accurate results.

Table 4 shows McNemar's test results by comparsion the proposed method with pervious methods. The McNemar's test results are $X^2 = 566.77$, $X^2 = 41.03$ and $X^2 = 594.72$ for Mnih (2013), Shu (2014) and Saito et al. (2016) compared with the proposed method. This means that the proposed method has produced significantly better results than other methods with a confidence level more than 99.9%.

#### 4.2.2. Abu Dhabi data

Results of applying the proposed method in Abu Dhabi data are shown in Fig. 7. The improvement by additional post-processing is obvious in building extraction than in the case of road extraction. The CNN map produces irregular building outlines and does not represent boundaries of adjacent buildings. It presents many adjacent buildings as one object, especially smaller buildings. In addition, there are some missing buildings. Compactness and density features of adjacent building regions help to smoothen outputs over some boundaries of predicted buildings because model uncertainty is quite higher for boundary parts ($Prob \geqslant T_{min}$); however, it does not solve the problem of separating adjacent buildings. This is because compactness and density of only adjacent regions with ($T_{min} \leqslant Prob \leqslant T_{max}$) are considered to define a building and increasing threshold leads to decrease in the overall performance. Also, increasing thresholds does not solve the problem of the gap between adjacent buildings.

In order to solve this problem, considering additional multi-scale input patch ($16 \times 16$, $32 \times 32$, $64 \times 64$, $96 \times 96$, $128 \times 128$, $160 \times 160$, $192 \times 192$, $224 \times 224$ and $256 \times 256$) could help to correct classification results of both coarse-scale and fine-scale detail in the image by considering the most common value. A similar concept was applied in Paisitkriangkrai et al. (2015). Another line of improvements is splitting building objects, regardless of their high probability to be building, by other discriminative features (e.g., morphological features). Another reason behind this problem is considering patch-based approach, which causes border discontinuities. Therefore, adding pixel-wise prediction or end to end prediction, instead of window-based approach, such as encoder-decoder or deconvolution using bi-linear interpolation to the full resolution (Kampffmeyer Michael and Jenssen, 2015; Maggiori et al., 0000), would enhance the final output. Also combination of contextual features from different stages of CNN can help

**Table 4**
McNemar's test in Massachusetts data.

| Methods | | Proposed method | | |
|---|---|---|---|---|
| | | Correct | Incorrect | Total |
| Mnih (2013) | Correct | 3,600,690 | 208,456 | **3,809,146 (18.5)** |
| | Incorrect | 768,716 | 15,957,028 | 16,725,744 (81.5) |
| | Total | **4,369,406 (21.3)** | 16,165,484 (78.7) | $X^2 = 566.77$ |
| Shu (2014) | Correct | 4,063,041 | 315,126 | **4,378,167 (21.3)** |
| | Incorrect | 348,549 | 15,808,174 | 16,156,723 (78.7) |
| | Total | **4,411,590 (21.5)** | 16,123,300 (78.5) | $X^2 = 41.03$ |
| Saito et al. (2016) | Correct | 3,608,083 | 201,518 | **3,809,601 (18.4)** |
| | Incorrect | 795,290 | 15,725,289 | 1,672,528 (81.4) |
| | Total | **4,403,373 (21.4)** | 16,131,517 (78.6) | $X^2 = 594.72$ |


**Fig. 7.** An example from Abu Dhabi data. (A) Test image, (B) ground-truth image, (C) the result obtained by convolutional network, (C) the result obtained after post-processing stage: red: road with high probability, green: building with high-probability, white: road/building with lower probability and considered as road/building. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

to improve results. This is because the first stage responds to low-level features such as edges and corners (fine spatial resolution); where each unit looks for small portion of input patch and the last stages respond to higher level of features such as objects (coarse spatial resolution); where each unit pools information from larger portions. Therefore, fine resolution feature map is necessary for localization and coarse-resolution feature map is important to differentiate between a flat rooftops and empty lots on the ground.

Table 5 presents a comparison between the proposed CNN with alternative CNN components in Abu Dhabi data. The proposed method outperforms other methods.

Table 6 summarizes the performance of each method at break-even point after applying the proposed post-processing step, where each class is trained separately as a binary classifier. It shows our single-class prediction outperforms others methods. However, all models have lower quality in extracting buildings. This is because buildings cover a large portion from input patch and Abu Dhabi map has higher spatial resolution, compared to Massachusetts map.

**Table 5**
Correctness (%) at breakeven of multi-class prediction in Abu Dhabi data.

| Dataset | Methods | Roads | Buildings |
|---|---|---|---|
| Abu Dhabi | Proposed method with GAP | 81.1 ± 2.0 | 78.8 ± 1.9 |
| | Proposed method with GMP | 78.2 ± 2.1 | 77.0 ± 2.2 |
| | Proposed method with FCLs | 79.8 ± 2.1 | 76.7 ± 2.1 |
| | Proposed method with GAP + Multi-scale | 81.5 ± 1.7 | 79.0 ± 1.8 |
| | Proposed method with GAP + CRF | 81.8 ± 1.9 | 80.2 ± 1.6 |
| | Proposed method with GAP + segmentation | 82.9 ± 1.7 | 81.6 ± 1.8 |

**Table 6**
Correctness (%) at breakeven in Abu Dhabi data by applying different methods.

| Dataset | Methods | Roads | Buildings |
|---|---|---|---|
| Abu Dhabi | Mnih (2013) | 78.3 | 75.9 |
| | Shu (2014) | 78.2 | 76.1 |
| | Saito et al. (2016) | 79.0 | 76.5 |
| | Proposed method | 80.9 | 77.9 |

**Table 7**
McNemar's test in Abu Dubai data.

| Methods | | Proposed method | | |
|---|---|---|---|---|
| | | Correct | Incorrect | Total |
| Mnih (2013) | Correct | 3,490,848 | 922,678 | **4,413,526 (7.2)** |
| | Incorrect | 1,617,667 | 55,573,477 | 57,191,144 (92.8) |
| | Total | **5,108,515 (8.3)** | 56,496,155 (91.7) | $X^2 = 436.04$ |
| Shu (2014) | Correct | 3,572,517 | 837,819 | **4,416,336 (7.2)** |
| | Incorrect | 1,456,645 | 55,737,689 | 57,194,334 (92.8) |
| | Total | **5,029,162 (8.2)** | 56575508 (91.8) | $X^2 = 408.53$ |
| Saito et al. (2016) | Correct | 3842803 | 1193852 | **5036655 (8.2)** |
| | Incorrect | 1267757 | 55300258 | 56568015 (91.8) |
| | Total | **5110560 (8.3)** | 56494110 (91.7) | $X^2 = 47.10$ |

Table 7 proves that the proposed method has superior performance because it is statistically significant with a confidence level more than 99.9% based on McNemar's test results $X^2 = 436.04$, $X^2 = 408.53$ and $X^=47.10$ for Mnih (2013), Shu (2014) and Saito et al. (2016), respectively.

Applying building rule (Eq. (9)) in the predicted building channel helps to reduce some FN cases where building superpixels are classified as background superpixels. Considering probability, compactness and density of adjacent superpixels to predicted building superpixels helps to reclassify some superpixels to buildings. Most of FN superixels, which achieve probability condition ($Prob \geqslant T_{min}$) and building condition ($(CI \geqslant T_{CI}) \cap (DI \geqslant T_{DI})$), are at boundaries of buildings and reclassified to buildings.

Similar to buildings, Applying road rule (Eq. (6)) in the predicted road channel reduces some misclassified cases by considering elongation and asymmetry ($(EI \geqslant T_{EI}) \cap (AI \geqslant T_{AI})$) in addition to probability of adjacent superpixels ($Prob \geqslant T_{min}$). Most of FN superpixels are between predicted road superpixels and consequently they are reclassified to road superpixels.

Although the proposed segmentation improves some cases, there are still some issues. The merging process using low-level features results in under-segmentation even if it removes the issue of many disjoint road regions. For instance, some road regions are partially blocked by trees and utilizing elongation and asymmetry properties of adjacent roads and tree segments can yield incomplete road regions. Merging superpixels based on similar shape features tends to produce over-segmentation even if it solves the problem of correctly segmenting some irregular large building segments. For example, it can result in the aggregation of two small building superpixels including pavement/driveway between them. Since the segmentation method assigns a class to a segment as a whole, there is no way to obtain accurate results in either case where the superpixels are incorrectly segmented. In general, patch-based method is based on low, middle and high-level features learned by CNNs to make a prediction pixel by pixel. Over/under segmentation can result in a classification result prone to errors in the presence of shadows and occlusions.

## 5. Conclusion

In this paper, a new method for extracting roads and buildings in high-resolution remote sensing imagery acquired over urban areas is proposed. The method does not consist of any pre-processing stage. It is based on patch-based CNN, which consists of five convolution layers and replaces fully connected layers with GAP. Low-level shape features of roads-regions (elongation and asymmetry) and building regions (compactness and density) are integrated with CNN features to connect disjointed road regions and complete irregular boundaries of building regions. The major contribution of this paper is the use of spatial features of adjacent superpixels to enhance multi-class prediction. The method is evaluated using two challenging datasets. The proposed method is also evaluated by employing different components of CNN architectures by replacing GAP with alternative components. It is also compared with other methods. The experiments prove that the method achieves good performance in the localization of urban objects, but it required additional processing to outline boundaries more accurately.

## Conflict of interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

## References


Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., Susstrunk, S., 2012. Slic superpixels compared to state-of-the-art superpixel methods. IEEE Trans. Pattern Anal. Mach. Intell. 34 (11), 2274–2282.

Badrinarayanan, V., Kendall, A., Cipolla, R., 2015. Segnet: a deep convolutional encoder-decoder architecture for image segmentation. Computing Research Repository (CRR) abs/1511.00561.

Bengio, Y., 2009. Learning deep architectures for ai. Found. Trend. Mach. Learn. 2 (1), 1–127.

Bengio, Y., 2012. Practical recommendations for gradient-based training of deep architectures. In: Neural Networks: Tricks of the Trade:. Second ed. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 437–478.

Brust, C.-A., Sickert, S., Simon, M., Rodner, E., Denzler, J., 2015. Convolutional patch networks with spatial prior for road detection and urban scene understanding. In: International Conference on Computer Vision Theory and Applications, pp. 510–517.

Brust, C.-A., Sickert, S., Simon, M., Rodner, E., Denzler, J., 2015. Efficient convolutional patch networks for scene understanding. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshop on Scene Understanding.

Castelluccio, M., Poggi, G., Sansone, C., Verdoliva, L., 2015. Land use classification in remote sensing images by convolutional neural networks. Computing Research Repository (CRR) abs/1508.00092.

Chen, L., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L., 2015. Semantic image segmentation with deep convolutional nets and fully connected crfs. In: International Conference on Learning Representations (ICLR), pp. 1–14.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., Fei-Fei, L., 2009. ImageNet: a large-scale hierarchical image database. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 248–255.

Felzenszwalb, P.F., Huttenlocher, D.P., 2004. Efficient graph-based image segmentation. Int. J. Comput. Vis. 59 (2), 167–181.

Firat, O., Can, G., Vural, F.T.Y., 2014. Representation learning for contextual object and region detection in remote sensing. In: 22nd International Conference on Pattern Recognition (ICPR), pp. 3708–3713.

Foody, G., 2004. Thematic map comparison: evaluating the statistical significance of differences in classification accuracy. Photogram. Eng. Rem. Sens. 70 (5), 627–633.

Hong, S., Noh, H., Han, B., 2015. Decoupled deep neural network for semi-supervised semantic segmentation. In: Neural Information Processing Systems (NIPS), pp. 1–9.

Hu, F., Xia, G.-S., Hu, J., Zhang, L., 2015. Transferring deep convolutional neural networks for the scene classification of high-resolution remote sensing imagery. Remote Sens. 7 (11), 14680.

Jabari, S., Zhang, Y., 2013. Very high resolution satellite image classification using fuzzy rule-based systems. Algorithms 6 (4), 762–781.

Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., Darrell, T., 2014. Caffe: Convolutional architecture for fast feature embedding.

In: Proceedings of the 22nd ACM International Conference on Multimedia, MM '14. ACM, pp. 675–678.
Jiang, Q., Cao, L., Cheng, M., Wang, C., Li, J., 2015. Deep neural networks-based vehicle detection in satellite images. In: International Symposium on Bioelectronics and Bioinformatics (ISBB), pp. 184–187.
Kampffmeyer Michael, A.-B.S., Jenssen, R., 2015. Semantic segmentation of small objects and modeling of uncertainty in urban remote sensing images using deep convolutional neural networks. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshop on Scene Understanding, pp. 1–9.
Kim, J., Lee, J.K., Lee, K.M., 2015. Accurate image super-resolution using very deep convolutional networks. Computing Research Repository (CRR) abs/1511.04587.
Krähenbühl, P., Koltun, V., 2011. Efficient inference in fully connected crfs with gaussian edge potentials. In: Advances in Neural Information Processing Systems 24. Curran Associates Inc., pp. 109–117.
Krizhevsky, A., Sutskever, I., Hinton, G.E., 2012. Imagenet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems 25. Curran Associates Inc., pp. 1097–1105.
Leeuw, J., Jia, H., Yang, L., Liu, X., Schmidt, K., Skidmore, A.K., 2006. Comparing accuracy assessments to infer superiority of image classification methods. Int. J. Rem. Sens. 27 (1), 223–232.
Lin, M., Chen, Q., Yan, S., 2014. Network in network. In: International Conference on Learning Representations, pp. 1–10.
Lin, G., Shen, C., Reid, I.D., van den Hengel, A., 2016. Efficient piecewise training of deep structured models for semantic segmentation. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR).
Liu, S., Deng, W., 2015. Very deep convolutional neural network based image classification using small training sample size. In: 3rd IAPR Asian Conference on Pattern Recognition (ACPR), pp. 730–734.
Liu, Z., Li, X., Luo, P., Loy, C.C., Tang, X., 2015. Semantic image segmentation via deep parsing network. In: IEEE International Conference on Computer Vision (ICCV), pp. 1377–1385.
Liu, B., Wu, H., Wang, Y., Liu, W., 2015. Main road extraction from zy-3 grayscale imagery based on directional mathematical morphology and vgi prior knowledge in urban areas. PLoS ONE 10 (9), 1–16.
Lngkvist, M., Kiselev, A., Alirezaie, M., Loutfi, A., 2016. Classification and segmentation of satellite orthoimagery using convolutional neural networks. Remote Sens. 8 (4), 329.
Long, J., Shelhamer, E., Darrell, T., 2015. Fully convolutional networks for semantic segmentation. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3431–3440.
Maggiori, E., Tarabalka, Y., Charpiat, G., Alliez, P., Fully convolutional neural networks for remote sensing image classification. IEEE International Geoscience and Remote Sensing Symposium (IGARSS).
Makantasis, K., Karantzalos, K., Doulamis, A., Doulamis, N., 2015. Deep supervised learning for hyperspectral data classification through convolutional neural networks. In: IEEE International Geoscience and Remote Sensing Symposium (IGARSS), pp. 4959–4962.
Marcu, A., Leordeanu, M., Dual local-global contextual pathways for recognition in aerial imagery. Computing Research Repository (CRR) abs/1605.05462.
Matinfar, H., Sarmadian, F., Panah, S.A., Heck, R., 2007. Comparisons of object-oriented and pixel-based classification of land use/land cover types based on lansadsat7, etm+ spectral bands (case study: arid region of iran). Am. Eurasian J. Agric. Environ. Sci. 2 (4), 448–456.
Maurya, R., Gupta, P.R., Shukla, A.S., 2011. Road extraction using k-means clustering and morphological operations. In: International Conference on Image Information Processing (ICIIP), pp. 1–6.
Mnih, V., 2013. Machine Learning for Aerial Image Labeling, Ph.D. thesis, University of Toronto.
Mnih, V., Hinton, G.E., 2010. Learning to detect roads in high-resolution aerial images. In: Proceedings of the 11th European Conference on Computer Vision (ECCV): Part VI, pp. 210–223.
Nair, V., Hinton, G.E., 2010. Rectified linear units improve restricted boltzmann machines. In: Proceedings of the 27th International Conference on Machine Learning (ICML-10). Omnipress, pp. 807–814.
Nogueira, K., Penatti, O.A.B., dos Santos, J.A., 2016. Towards better exploiting convolutional neural networks for remote sensing scene classification. Pattern Recogn. abs/1602.01517.
Noh, H., Hong, S., Han, B., 2015. Learning deconvolution network for semantic segmentation. In: IEEE International Conference on Computer Vision (ICCV), pp. 1520–1528.
Oquab, M., Bottou, L., Laptev, I., Sivic, J., 2015. Is object localization for free? - weakly-supervised learning with convolutional neural networks. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 685–694.
Paisitkriangkrai, S., Sherrah, J., Janney, P., Hengel, A.V.-D., 2015. Effective semantic pixel labelling with convolutional networks and conditional random fields. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, pp. 36–43.
Papandreou, G., Chen, L.C., Murphy, K.P., Yuille, A.L., 2015. Weakly-and semi-supervised learning of a deep convolutional network for semantic image segmentation. In: IEEE International Conference on Computer Vision (ICCV), pp. 1742–1750.
Rajeswari, M., Gurumurthy, K.S., Omkar, S.N., Senthilnath, J., Reddy, L.P., 2011. Automatic road extraction using high resolution satellite images based on level set and mean shift methods. In: 3rd International Conference on Electronics Computer Technology (ICECT), vol. 2, pp. 424–428.
Saito, S., Aoki, Y., 2015. Building and road detection from large aerial imagery. In: Proceedings of Society of Photographic Instrumentation Engineers (SPIE) - The International Society of Optical Engineering, vol. 9405.
Saito, S., Yamashita, T., Aoki, Y., 2016. Multiple object extraction from aerial imagery with convolutional neural networks. J. Imag. Sci. Technol. 60 (1). 010402-1–010402-9.
Sherrah, J., 2016. Fully convolutional networks for dense semantic labelling of high-resolution aerial imagery. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR).
Shu, Y., 2014. Deep Convolutional Neural Networks for Object Extraction from High Spatial Resolution Remotely Sensed Imagery, Ph.D. thesis, University of Waterloo.
Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R., 2014. Dropout: a simple way to prevent neural networks from overfitting. J. Mach. Learn. Res. 15 (1), 1929–1958.
Sujatha, C., Selvathi, D., 2015. Connected component-based technique for automatic extraction of road centerline in high resolution satellite images. EURASIP J. Image Video Process. 15 (2008), 1–16.
Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A., 2015. Going deeper with convolutions. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1–9.
Tremeau, A., Colantoni, P., 2000. Regions adjacency graph applied to color image segmentation. IEEE Trans. Image Process. 9 (4), 735–744.
Vakalopoulou, M., Karantzalos, K., Komodakis, N., Paragios, N., 2015. Building detection in very high resolution multispectral data with deep learning features. In: IEEE International Geoscience and Remote Sensing Symposium (IGARSS), pp. 1873–1876.
Visin, F., Kastner, K., Courville, A.C., Bengio, Y., Matteucci, M., Cho, K., 2015. Reseg: a recurrent neural network for object segmentation. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR).
Yu, H., Yang, W., Xia, G.-S., Liu, G., 2016. A color-texture-structure descriptor for high-resolution satellite image classification. Remote Sens. 8 (3), 259.
Zeiler, M.D., Fergus, R., 2014. Visualizing and understanding convolutional networks. In: European Conference on Computer Vision (ECCV), pp. 818–833.
Zhao, R., Ouyang, W., Li, H., Wang, X., 2015. Saliency detection by multi-context deep learning. In: IEEE conference on Computer Vision and Pattern Recognition (CVPR).
Zheng, S., Jayasumana, S., Romera-Paredes, B., Vineet, V., Su, Z., Du, D., Huang, C., Torr, P.H.S., 2015. Conditional random fields as recurrent neural networks. In: IEEE International Conference on Computer Vision (ICCV), pp. 1529–1537.
Zhou, B., Lapedriza, A., Xiao, J., Torralba, A., Oliva, A., 2014. Learning deep features for scene recognition using places database. In: Advances in Neural Information Processing Systems 27, pp. 487–495.
Zhou, B., Khosla, A., Lapedriza, À., Oliva, A., Torralba, A., 2015. Object detectors emerge in deep scene cnns. In: International Conference on Learning Representations (ICLR).
Zhou, B., Khosla, A., Lapedriza, À., Oliva, A., Torralba, A., 2015. Learning deep features for discriminative localization. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR).