ARTICLE

Automated LULC Map Production using Deep Neural Networks

Christopher J. Henry[a], Christopher Storie[b], Muthu Palaniappan[ca], Victor Alhassan[a], Mallikarjun Swamy[ad], Damilola Aleshinloye[a], Andrew Curtis[b], and Daeyoun Kim[a]

[a]University of Winnipeg, Department of Applied Computer Science, 515 Portage Avenue, Winnipeg, Manitoba, Canada;[b]University of Winnipeg, Department of Geography, 515 Portage Avenue, Winnipeg, Manitoba, Canada;[c]College of Engineering Guindy, Department of Information Science and Technology, Chennai, India;[d]Birla Institute of Technology and Science, Pilani, Pilani Campus, Rajasthan, India

ARTICLE HISTORY
Compiled December 21, 2018

ABSTRACT
This article presents an approach to automating the creation of land-use/land-cover classification (LULC) maps from satellite images using deep neural networks that were developed to perform semantic segmentation of natural images. This work is important since the production of accurate and timely LULC maps is becoming essential to government and private companies that rely on them for large-scale monitoring of land resource changes. In this work, deep neural networks re trained to classify each pixel of a satellite image into one of a number of LULC classes. The presented deep neural networks are all pre-trained using the ImageNet Large-Scale Visual Recognition Competition (ILSVRC) datasets and then fine-tuned using approximately 19,000 Landsat 5/7 satellite images of resolution 224 × 224 taken of the Province of Manitoba in Canada. The result is an automated solution that can produce LULC maps images significantly faster than current semi-automated methods. The contributions of this article are the observation that deep neural networks developed for semantic segmentation can be used to automate the task of producing LULC maps; the use of these networks to produce LULC maps; and a comparison of several popular semantic segmentation architectures for solving the problem of automated LULC map production.

KEYWORDS
Land-use/land-classification; deep neural networks; fully convolutional networks; semantic segmentation

1. Introduction

The creation of land-use/land-cover classification (LULC) maps is one of the most common applications of remote sensing. Due to decreasing cost and increasing resolution of satellite imagery, many government and private industries are turning to LULC maps as an important tool for large-scale monitoring of land resource changes. These maps are vital in areas such as flood forecasting, urban and rural land-use planning, resource management, and disaster management and planning (Treitz and Rogan 2004). Consequently, the production of accurate and timely maps is becoming increasingly important. Current approaches to producing LULC maps are typically

CONTACT C. J. Henry. Email: ch.henry@uwinnipeg.ca

semi-supervised (per pixel or object based) and require considerable user input to ensure high classification accuracies. These semi-automated approaches can be prone to error, take a significant amount of training and classification time, and suffer from a lack of consistency – especially when multiple analysts are involved in the process and/or when dealing with large multi-scene areas. Moreover, it is difficult to improve accuracy and efficiency of these methods as they typically rely on pixel pattern matching and manual user input. As a result, the problem considered in this paper is the use of deep neural networks for producing land-use/land classification (LULC) maps from satellite images to alleviate the aforementioned problems.

In the proposed process, each pixel in a satellite image must be classified into a number of land-use classes (*e.g.* deciduous forest, marshland, fens, *etc.*), and the proposed solution is only possible since automated systems based on machine learning methods (Domingos 2015), called deep learning algorithms (Goodfellow, Bengio, and Courville 2016), have advanced to a point where they can be applied to real-world problems (LeCun, Bengio, and Hinton 2015). These systems excel by performing pattern classification tasks humans find repetitive and tedious in nature – at significantly faster speeds, have been shown to surpass human performance in some instances (He et al. 2015b), and, once trained, produce consistent results. Recently, these networks have been used to perform semantic segmentation of images (Long, Shelhamer, and Darrell 2015; Shelhamer, Long, and Darrell 2017). This is the problem of partitioning a digital image into a number of segments, where each segment captures some perceptual object within the images. These networks are trained to make decisions on pixel classes based on pixel feature values, local structure and texture, and high-level perceptual content in the image. Thus, a principal contribution of this paper is the observation that the problem of semantic segmentation of digital images directly links to the challenge of classifying satellite image pixels into LULC categories.

The solution presented here is based on the fully convolutional network (FCN), introduced by Long, Shelhamer, and Darrell (2015); Shelhamer, Long, and Darrell (2017), that is trained to classify every pixel in a satellite image into one of a number of land-use class. This work was a result of a pilot project funded by GeoManitoba, which is part of the Ministry of Sustainable Development within the province of Manitoba. The goal of the project was to investigate whether machine learning methods could reduce the amount of time required to produced LULC maps. GeoManitoba is mandated to produce LULC maps of Manitoba for various land resource planning activities by both government and private entities. For this reason, they have many years worth of LULC maps of the Province of Manitoba generated from Landsat 5/7 data. While the lack of large labelled datasets are typically a barrier to applying deep neural networks to new problem domains, GeoManitoba's maps were used to create a labelled dataset vital for training neural networks. The results contained in this article were generated from this dataset. This article is an extension of the work reported by Storie and Henry (2018), and the contributions of this article are: 1) the observation that deep neural networks developed for semantic segmentation can be used to automate the task of producing LULC maps; 2) the use of FCN to produce LULC maps; and 3) a comparison of this approach to conditional random field as a recurrent neural networks (CRF-RNN) (Zheng et al. 2015) and multi-scale context aggregation by dilated convolutions (Yu and Koltun 2015). This work represents a natural application of deep learning neural networks, that were developed for performing semantic segmentation of natural colour images, to remote sensing and geoanalytics.

2. Deep Leaning Applications to Remote Sensing

This section provides background information on deep learning developments that made the presented solution possible and its recent use in the field of remote sensing.

2.1. *Deep Neural Networks*

Since 2012 there has been a surge in applications of deep learning. These developments were only possible due to the confluence of general purpose computing using graphics processing units (GPUs) (Kirk and Hwu 2017), large labelled datasets (Russakovsky et al. 2014), and the introduction of deep neural networks, which contain many more layers and parameters than traditional neural networks (LeCun, Bengio, and Hinton 2015; Goodfellow, Bengio, and Courville 2016). In particular, Krizhevsky, Sutskever, and Hinton (2012) created a deep learning model (AlexNet) used for image processing tasks. AlexNet won the 2012 ImageNet Large-Scale Visual Recognition Competition (ILSVRC) - by a significant margin - and largely popularized deep convolutional neural network (DCNNs). Briefly, a neural network is a mathematical abstraction based (very) loosely on the behaviour of neurons in the brain. A neural network is composed of layers of neurons, and the term deep learning characterizes a neural network with many more layers and many more parameters than a traditional neural network. These algorithms "learn" to perform tasks by extracting patterns from large labelled datasets. They are exposed to an example, and the right answer for a given task is used to tune the network. Through repeated training on large datasets, the network is able to extract patterns and perform the correct action for a given input. The important observation here is that a human has not explicitly programmed the system behaviour. Instead, the algorithm "learns" during training based on labelled data. Thus, all the impressive results achieved with deep neural networks are data driven through the use of big-data datasets. Since 2012, DCNN have been successfully applied in many interesting applications to produce amazing results, such as robotics (Levine et al. 2016), speech recognition (Bahdanau et al. 2015), and natural language processing (Cho et al. 2014).

DCNN builds on the insight of LeNet created by LeCun et al. (1998), which is widely recognized as the first convolutional neural network (CNN). Wang, Raj, and Xing (2017) provide resourceful insight on the origin of deep learning models. A typical DCNN model comprises of convolution layers, pooling layers, activation layers, and fully-connected layers. AlexNet consisted of five convolutional and max-pooling layers, and three fully-connected layers with a final 1000-way softmax (Li and Karpathy 2015) that completes the model. Training AlexNet was only possible in a practical amount due to the use of GPUs. In particular, the parallel structure of DCNNs make them a perfect candidate for GPU-based acceleration. It is for this reason that GPUs play a significant role in the practical implementation of DCNN (Chetlur et al. 2014). Subsequent DCNNs models increased the depth of CNNs, and introduced new techniques which improved training efficiency and accuracy of the models. See (LeCun, Bengio, and Hinton 2015; Zhu et al. 2017) for a complete history of DCNN and reviews of DCNNs models VGG (Simonyan and Zisserman 2014), ResNet (He et al. 2015a) and fully convolutional networks (Long, Shelhamer, and Darrell 2015). In recent times, DCNNs have been successfully adapted to solve remote sensing problems. Examples of seminal work in this domain are given in the next subsection.

2.2. *Deep Neural Networks Related Work in Remote Sensing*

Castelluccio et al. (2015) explored the use of DCNNs for classification of remote sensing scenes using two contemporary architectures, namely CaffeNet (a modified version of AlexNet) and GoogleNet (Szegedy et al. 2014). Moreover, they considered two datasets, UC-Merced (21 classes) (Yang and Newsam 2010) and the Brazilian Coffee Scenes (4 classes) (Penatti, Nogueira, and dos Santos 2015), since each dataset contained unique feature. UC-Merced spatial and spectral characteristics (high resolution, low level features, and RGB colour space) were closely matched to general optical images, while the Brazilian Coffee Scenes includes a special near infrared band typically found in remote-sensing data. CaffeNet and GoogleNet were implemented to classify both datasets independent of each other. Results from both models outperformed other classical techniques paired. Although the results are impressive, this approach discards spatial location and context information. Moreover, objects recognized in the input data are classified by assigning label classes in form of probabilities, which means low and high-level features are not visually represented.

Basu et al. (2015) proposed a novel classification framework for classifying satellite images called DeepSat. 150 features were extracted from two datasets (SAT-4 and SAT-6) containing four bands (red, green, blue and near infrared), and some features extracted include energy, entropy, homogeneity, contrast, maximum probability, saturation, intensity, and image channels. All features were normalized to lie in the range [0, 1] before being fed to a deep belief network classifier trained using the contrastive divergence algorithm (Carreira-Perpiñán and Hinton 2005). This network outperforms the classical deep belief network classifier on the target dataset. However, this approach is evaluated on high-resolution aerial imagery (1-m) containing four bands hence the features considered are data dependent and makes it difficult to analyze the performance of this approach for low-resolution satellite imagery(30-m) containing six bands addressed in our work. Furthermore, this approach discards spatial location and context information and only classifies high-order abstraction objects.

Marmanis et al. (2016) explore a system for classification of remote sensing images by extracting representations from DCNNs that are pre-trained on ImageNet dataset. The system follows a two stage classification scheme. The first stage feeds original training data into a pre-trained DCNN model to extract information from activations in the last layers of the pre-trained DCNN. This data is then fused to a single vector reshaped into a 2-D array. The second stage uses this information to train a CNN classifier supervised. This system has three positive implications: richer information obtained in the deeper layers of pre-trained networks contributes to higher classification accuracy; information fusion from different layers influences accuracy since multiple scales of relevant information exists at these layers; and, by reshaping the single vector into 2-D array, a reduction of parameters occurs resulting in improved processing of features by the CNN classifier. This approach is similar to transfer learning and fine-tunning of DCNNs (Yosinski et al. 2014a) on a target dataset for a task. This is an effective way of training DCNNs when constrained with small datasets, and is an approach employed in the work presented here. Although the results documented by the researchers are impressive, this method did not consider satellite data with greater spectral resolution and geographical variations.

Zhang, Zhang, and Kumar (2016) compiled a technical tutorial on the state of art in deep learning techniques for remote sensing. Similarly, Zhu et al. (2017) present a comprehensive review of deep learning in remote sensing coupled with a detailed list of resources. Also, Ball, Anderson, and Chan (2017) present a great survey of deep learning in remote sensing by highlighting challenges and open problems of deep learning in remote sensing while introducing tools and theories to this community.

Lastly, this section concludes with work related to the problem considered in this paper. Zhao et al. (2016) use Conditional Random Fields (CRFs) to perform the same task as reported here. We choose to use the work by Zheng et al. (2015) since they formulated a CRF as a Recurrent Neural Network (RNN) which allows the solution to be trained end-to-end using back-propagation. Han, Zhong, and Zhang (2016) use unsupervised convolutional sparse auto-encoders for spatial-spectral classification. This paper considers hyperspectral imagery which can have 30-100s of individual bands with considerable redundancy between t.hem For example the red band in Landsat covers a spectral range of 1 nm, however within some hyperspectral systems this may be covered by 5 individual bands. Hyperspectral systems are typically used in situations where a particular object (such as a rock) is sensitive to very specific wavelengths that allow it to be differentiated from other similar objects. The challenge is to reduce the data redundancy to only those specific useful bands. Secondly, hyperspectral imaging is not widely used in LULC mapping of large geographic areas where there are broad, generalized LULC categories as there is an over collection of data and the ability to distinguish between broad classes is much easier. Moreover, their approach is unsupervised, while the data in this work was labelled by GeoManitoba. Finally, Zhao et al. (2018) use a game-theoretic spectral-spatial classification algorithm using a CRF model. Again, their approach is based on hyperspectral imagery, which is different than the data used in this work. In particular, the authors were trying to distinguish between much more highly related vegetation covers which benefit from hyperspectral data.

3. Semantic Segmentation Using Deep Neural Networks

The solution presented in this paper is based on neural networks developed to perform semantic segmentation of digital images. This section presents the different semantic segmentation neural networks used to generate the results in this paper. All of the presented methods stem from the work by Jonathan Long, Evan Shelhamer, and Trevor Darrell in using fully CNNs for performing semantic segmentation of images, and subsequent contributions their work inspired (see below). Semantic segmentation is the process of partitioning an image such that each pixel in the image is assigned a unique label corresponding to the perceptual content within the image. An example of semantic segmentation is given in Fig. 1. Semantic Segmentation using deep neural networks is one of the many problems that has gained significant traction lately. This problem is considerably difficult - and different from problems such as image classification, face recognition, *etc.* - since it involves pixel-level prediction. Nevertheless, a lot of effort has been put into adapting the existing neural network architectures to solve this problem. To this end, the following delves into some of the important semantic segmentation architectures.

CNNs are a variant of neural networks which were specifically designed for the purpose of computer vision tasks. The basic building blocks of CNN architectures are layers which are capable of performing various functions. Broadly speaking, as the input image progresses through these layers, either a filter is applied upon it or it is sub-sampled to a smaller size. These operations finally lead to a fully connected layer, where each neuron has connections to all of the output of previous layer. This layer facilitates the classification task. The entire set of operations compute a general non-

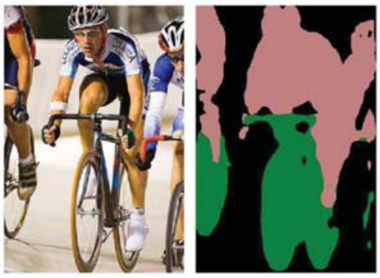


Figure 1. Example of semantic segmentation (Zheng et al. 2015).

linear function. If, however, the fully connected layer is re-interpreted as a convolution layer (*i.e.* as a filter), then the CNN is called a deep filter or a fully convolutional network (FCN) (Long, Shelhamer, and Darrell 2015; Shelhamer, Long, and Darrell 2017). This modification is rather trivial and produces an output of re-sampled spatial dimension (*i.e.* spatially reduced in size). FCN forms the basis of the model that Long, Shelhamer, and Darrell (2015) propose and is used to produce the results presented in this paper.

To address the fact that the FCN produces sub-sampled output, the FCN is appended with deconvolution layers (Dumoulin and Visin 2016) which compensate for the lost spatial information by up-sampling. The weights for the these layers can also be learned using back-propagation. Additionally, a skip architecture is introduced by Long *et al.* to enhance the coarse semantic information produced by traditional convolutional layers with shallow, fine, appearance information. This new network architecture, consisting of the original deep network and the up-sampling layers, is called an FCN and it is end-to-end trainable (Long, Shelhamer, and Darrell 2015; Shelhamer, Long, and Darrell 2017). The original network used by Long, Shelhamer, and Darrell (2015) to define the FCN was the VGG (Simonyan and Zisserman 2014) network. The same approach was used for this article.

The work by Long, Shelhamer, and Darrell (2015) was seminal in regards to the problem of semantic segmentation. However, there were several extensions of the FCN that were also reported at the beginning of the work reported here. As a result, several methods were considered in order to find the best solution for the problem of automating LULC map production. The first of these methods (after the FCN) was conditional random field as a recurrent neural networks (CRF-RNN) (Zheng et al. 2015). CRF-RNNs were developed on the premise that good feature representations are needed to produce the best results for a semantic segmentation task. Specifically, the course output due to the large receptive field of convolution and the max pooling layers reduce the global semantic information available in the network. Probabilistic graphical models such as conditional random fields (CRFs) have proven to be successful as a post processing technique for semantic segmentation. Building upon this, Arnab et al. (2015); Zheng et al. (2015), utilize the capacity of CRFs by implementing them in the form of a recurrent neural network (RNN) which is appended to the FCN. This modification leads to yet another network that can be trained end-to-end using back-propagation .

The other method used for comparison with FCN is multi-scale context aggregation by dilated convolutions (Yu and Koltun 2015). Deep learning models for semantic segmentation are based on adaptations of convolutional networks that had originally been designed for image classification. However, dense prediction problems such as semantic segmentation are structurally different from image classification. Yu and Koltun (2015) proposes a new convolutional network module that is specifically designed for dense prediction by removing vestigial components and introducing a context module. The presented module uses dilated convolutions to systematically aggregate multi-scale contextual information without losing resolution. The architecture is based on the fact that dilated convolutions support an exponential expansion of the receptive field and linear parameter accretion without loss of resolution or coverage.

The front-end of this network is adapted from VGG-16 network (Simonyan and Zisserman 2014) for dense prediction by removing the last two pooling and striding layers. Furthermore, convolutions in all subsequent layers were dilated by a factor of 2 for each pooling layer that was ablated. These vestiges were considered to be counter-productive for dense prediction. This network is labelled *Dilation 8 Frontend* (D8 Frontend). Next, the context module was designed to increase the performance of dense prediction architectures by aggregating multi-scale contextual information. The basic context module has 7 layers that apply 3×3 convolutions with increasing dilation factors as 1, 1, 2, 4, 8, 16, and 1. Each of these convolutions is followed by a pointwise truncation $\max(\cdot, 0)$. A final layer performs $1 \times 1 \times C$ convolutions and produces the output of the module, where C is the number of classes. Modern semantic segmentation networks integrate multi-scale contextual information via successive pooling and sub-sampling layers that reduce resolution until a global prediction is obtained. The dilated network solves this problem by exponentially increasing the receptive field without down-scaling the image severely. This network is labelled *D8 Context*.

Finally, there are many other FCN extensions that were not considered in this work, but are related works (Shelhamer, Long, and Darrell 2017). Pathak, Krahenbuhl, and Darrell (2015) introduce an approach to semantic labelling by constraining FCNs to a latent distribution of ground-truth pixel labels. In this weakly supervised setting, the network is presented with training samples paired with image level tags obtained from label classes present in the ground-truth, a combination of constraint settings are used to extract the image level labels. Network output is produced by optimization of a loss function to closely follow the latent distribution. Papandreou et al. (2015) incorporates expectation-maximization algorithm with DCNNs for semantic segmentation in the weakly supervised setting. The algorithm estimates latent pixel labels while optimizing the DCNNs parameters using stochastic gradient descent. Dai, He, and Sun (2015) investigate the use of bounding box annotations independently or as additional source of supervision to train FCNs for semantic segmentation. Segmentation masks are estimated from the bounding boxes using region proposal methods. Hong, Noh, and Han (2015) perform semantic segmentation with decoupled DCNNs. This approach decouples a DCNN into classification and segmentation space, each independent of the other, and learns from a large number of weak image level annotations and a few dense-pixel labels. Bridging layers are introduced between the two network spaces, which delivers class-specific information from the classification space to the segmentation space. Semantic labelling is achieved by optimizing two separate objective functions for each space while both networks collaborate by sharing information. This method aims to train DCNNs in the weakly supervised settings where availability of dense-pixel labels is limited.

4. Network Modifications

This section presents the modifications to the networks given in Section 3 that were necessary to produce the results reported below. In order to discuss these changes, this section begins with a brief review of a convolutional neural network layer using the notation in (Li and Karpathy 2015). A convolutional layer consists of a rectangular volume of neurons as depicted in Fig. 2. A depth slice (depicted as the black rectangle in Fig. 2) is a rectangular array of neurons taken from the volume. Each layer is characterized by its depth (K), stride (S), zero-padding (P), receptor field size (F), and input volume dimension (W). Each neuron in the volume uses a receptor field of values from the previous layer to calculate its output. This requires a set of weights equal to the size of the receptor field multiplied by the input depth. In order to reduce the computational costs of a convolutional layer, the neurons in each depth slice share a set of weights and a bias. Thus, the number of parameters (*i.e.* weights) required for each volume is equal to its depth × receptor field area × input depth + one bias for each depth slice, *i.e.* $K \times F^2 \times$ input depth $+ K$. For instance, the original VGG network operated on RGB images and set $F = 3$. Thus, the input depth to the Conv1.1 layer was 3 and the number of parameters for this layer is $64 \times 3^2 \times 3 + 64$ =1,792. Furthermore, the convolutional layer output dimension is calculated as $(W - F + 2P)/S + 1$. Similarly, for the VGG network $P = S = 1$, which gives the output dimension for Conv1.1 is $(224 - 3 + 2)/1 + 1 = 224$.

As mentioned in Section 3, the three networks considered in this paper are FCN, CRF-RNN, and D8 (both frontend and context). These networks were almost unaltered from their original publications. The only change required in this work is due to the fact that Landsat 5/7 satellite images contain six bands (red, green, blue, and three infrared channels) instead of the RGB-based images used in the original networks. All the networks reported in Section 3 are based on convolutional neural networks. As was mentioned, in these structures, neurons are grouped into three-dimensional volumes, and these volumes are grouped together to form layers. For example, the input and output dimensions of each layer in the FCN network used in this work are given in Table 1. Notice, that the base of this network is almost identical to the VGG (Simonyan and Zisserman 2014) network. The only difference is that the last FC-1000 layer has been removed and three deconvolutional layers have been added. This specific structure corresponds to the FCN-8 network which is described in great detail by Long, Shelhamer, and Darrell (2015); Shelhamer, Long, and Darrell (2017). In the case of satellite images, the first layer (*i.e.* the input image) becomes 224 × 224 × 6, which means that the input to the first convolutional layer is doubled from 224 × 224 × 3. This is handled in our networks by doubling the number of parameters in this first layer (see, *e.g.* Fig. 3). Note, the number of neurons does not increase because the layer dimensions remain $224 \times 224 \times 64$), only the number the weights for this layer doubles since $64 \times 3^2 \times 6 + 64$ =3,520. Furthermore, both randomly initializing these weights and copying the existing pre-trained weights were considered in this work, with no observable differences. As a result, all the weights in the new layers were initialized with random values for all the results presented in this paper.

5. Dataset Background and Pre-Processing

In this work, Landsat 5/7 data was used, where each pixel in a satellite image corresponds to a 30 m × 30 m square area of land. Moreover, each pixel is associated

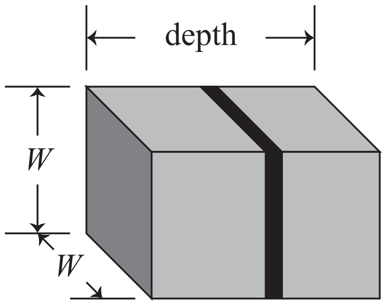
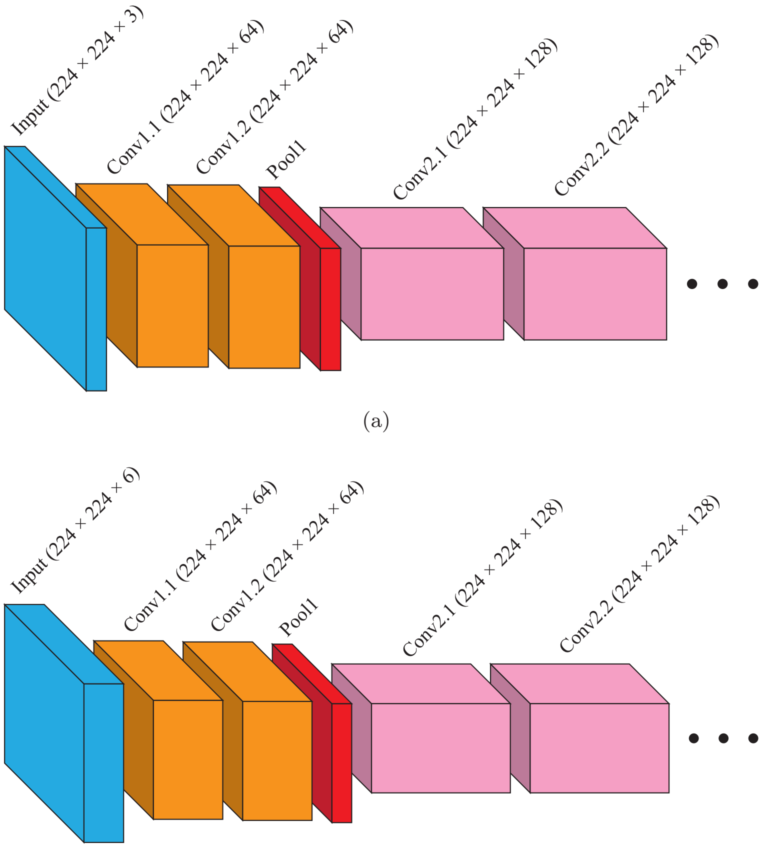


Figure 2. Example of a convolutional layer organized into a volume of neurons, where the black area represents a depth slice of neurons (Li and Karpathy 2015).



(a)

(b)

Figure 3. Example of network modifications performed in this work. (a) First six layers of the FCN-8 using RGB image input (*i.e.* 3 bands), and (b) first six layers of the FCN-8 with Landsat 5/7 (*i.e.* 6 bands).

with a tuple of values containing three infrared channels in addition to the typical RGB channels found in most digital images. The labelled dataset was created using existing Landsat 5/7-based LULC maps produced by GeoManitoba. The maps provided by GeoManitoba are for the southern agricultural growing region of Manitoba (see the red outline in Fig. 4), which will be referred to as the *southern extent of Manitoba*. The size of this region is approximately 148,800 km^2. These maps were created using semi-automated methods (see, *e.g.* Ban, Gong, and Giri (2015)), and were then ground-truthed. A total of eighteen Landsat 5/7 scenes (see the green outlines in

Table 1. Input/output characteristics of the FCN-8 network used in this work.

Input				**Output**		
Layer	Height	Width	Depth	Height	Width	Depth
Input Image				224	224	6
Conv1.1	224	224	6	224	224	64
Conv1.2	224	224	64	224	224	64
Pool1	224	224	64	112	112	64
Conv2.1	112	112	64	112	112	128
Conv2.2	112	112	128	112	112	128
Pool2	112	112	128	56	56	128
Conv3.1	56	56	128	56	56	256
Conv3.2	56	56	256	56	56	256
Conv3.3	56	56	256	56	56	256
Pool3	56	56	256	28	28	256
Conv4.1	28	28	256	28	28	512
Conv4.2	28	28	512	28	28	512
Conv4.3	28	28	512	28	28	512
Pool4	28	28	512	14	14	512
Conv5.1	14	14	512	14	14	512
Conv5.2	14	14	512	14	14	512
Conv5.3	14	14	512	14	14	512
Pool5	14	14	512	7	7	512
FC6	7	7	512	7	7	4096
FC7	7	7	4096	7	7	4096
FC-Final	7	7	4096	7	7	19
Deconv1	7	7	19	14	14	512
Deconv2	14	14	512	28	28	256
Deconv3	28	28	256	224	224	19

Fig. 4) were used to produce the maps. LULC maps are produced from satellite images by classifying each pixel in the satellite image to one of several land-use labels (see, *e.g.*, Fig. 5). Example labels include water, grassland, marsh, deciduous, coniferous, road, and agriculture. The full list is given in Fig. 6, and an example of a GeoManitoba LULC map created from Landsat 5/7 data from 2004 is given in Fig. 7. LULC maps have many applications, including flood forecasting, urban and rural land-use planning, resource management, and disaster management and planning. GeoManitoba is mandated to create provincial land-use maps on a regular basis to assist in these activities. GeoManitoba estimates that the time to create a single LULC map using semi-automated methods is approximately 4,800 hours (or 600 work days). However, due to limits on personnel from budget restraints, and the ability to obtain relatively cloud free imagery, the process of creating LULC maps for the southern extent of Manitoba can take as long as 2-3 years to produce and can incur significant labour costs in the process. Finally, as has been mentioned, a network must provide a label for each pixel. As a result, the GeoManitoba dataset was augmented with satellite images containing clouds and a new class was added to the list provided by GeoManitoba. Without this addition, the final system would always misclassify clouds into one of the other land-use classes.

The LULC maps provided by GeoManitoba are produced using imagery acquired from mid-May to late August. The dates correspond to the normal growing season and also represent a period where most vegetation is in bloom. By using this range of dates, any seasonality effects are minimized. Secondly, there was no effort to minimize radiation differences as they are natural and cannot be predicted. By not controlling radiation differences the neural network is better able to learn the various impacts the radiation differences will have on the particular LULC category and better able to classify future images into the correct category. Lastly, for the purposes of this research the neural network was trained using 8-bit data and consequently can only classify 8-bit data. Future work is underway to allow for the use of Landsat 8, 16-bit data.

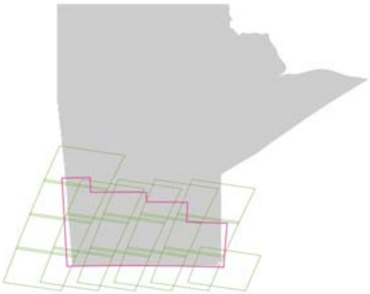
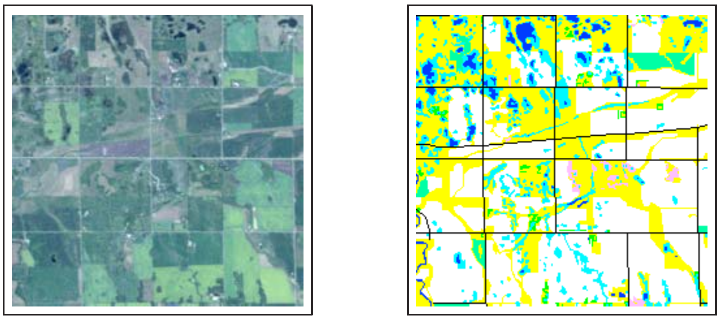


Figure 4. Province of Manitoba with the southern agricultural growing region (red) and the associated Landsat 5/7 scenes that cover this area (green).



(a)

(b)

Figure 5. Example LULC map. (a) RGB components of a Landsat 7 satellite image of Manitoba, and (b) the GeoManitoba LULC map produced from (a). Note, each pixel has a dimension of 30 m × 30m and, since each tile is of size 224 × 224, the images in (a) & (b) is have dimensions of 6.72 km × 6.72 km.

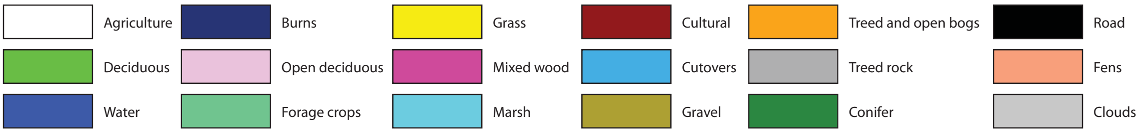


Figure 6. GeoManitoba land-use classes

6. Data requirements

The deep neural networks used in this work were all originally designed around a specific input image resolution. For example, the FCN network used images of size 224 × 224. To produce a working solution, the original input image resolution for these networks was not modified. In other words, the input resolution of the candidate networks was not increased to the size of the southern extent of Manitoba. This decision was made based on the following observations. First, these networks require a very large number of images to train. For example, all the presented solutions are based on the VGG network that was developed for the ILSVRC (Russakovsky et al. 2014), where the training set consisted of approximately 1.2 million images and 1000

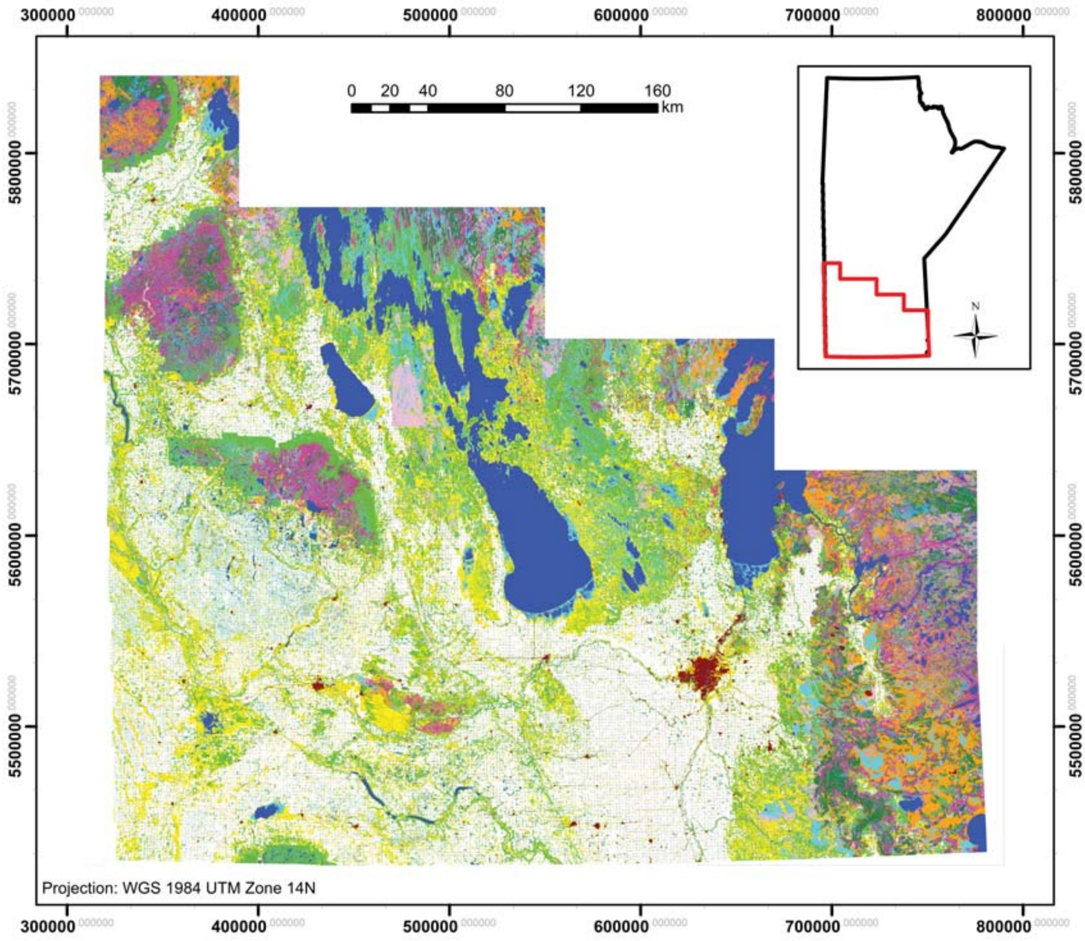


Figure 7. 2004 LULC map provided by GeoManitoba.

categories. Keeping the full resolution of the LULC maps produced by GeoManitoba would mean that only three training images were available. Furthermore, despite the methods reported in this section, there was simply not sufficient data to train a full network from scratch. As a result, transfer learning techniques (also called fine-tuning) were employed for all the networks (Yosinski et al. 2014b). Transfer learning consists of adapting a deep neural network that has been pre-trained on the ILSVRC problem by training with a smaller dataset for a related – but different – problem. By maintaining the original input resolution for these networks, transfer learning was employed and the images (both the raw satellite data and associated LULC maps) were divided into tiles that dramatically increased the number of images (and corresponding LULC map labels) available for training. In particular, for the area shown in Fig. 7, this approach produced 19,012 images of size 224 × 224 for this work, which were divided into training and validation sets of size 18,054 and 958, respectfully. The process used to produce the individual tiles from the full southern extent is depicted in Fig. 8. The first set of tiles were produced by the method shown in Fig. 8(b), namely the tiles were non-overlapping. To further increase the size of the dataset, the tiling process in Fig. 8(c) was used to produce more tiles. In this case, tiles were overlapped by half the size of the network input resolution, *i.e.* $224/2 = 112$. Furthermore, by starting this process in each of the four corners of the full map depicted in Fig. 8(a), the total number of tiles generated in Fig. 4 (b) was able to be increased by more than 4× due to the fact that the resolution of the map in Fig. 8(a) is not a multiple of 224.

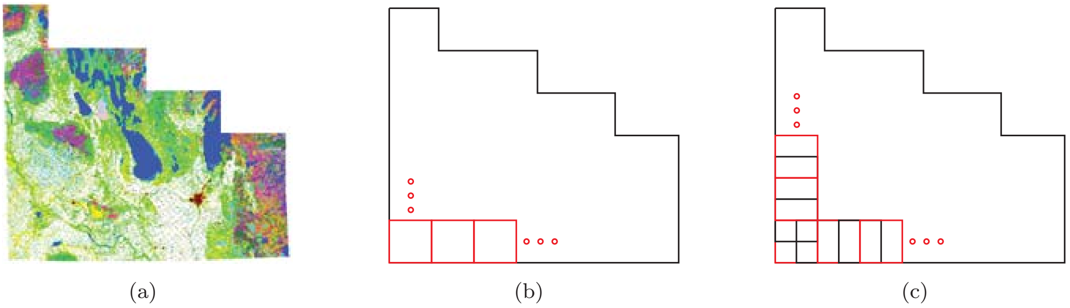


Figure 8. Illustration depicting the tiling process. (a) GeoManitoba's LULC map of southern Manitoba, (b) non-overlapping tiles, and (c) 1/2 tile overlap.

7. Experiment, Results, and Analysis

This section presents the accuracy of the trained networks and samples of the output LULC maps. The networks identified in Section 3 were trained with the dataset described in Section 6. The results of average accuracy of the network mentioned above are reported in Table 2, which was a comparison of the networks to find the solution that performed best on the GeoManitoba dataset. These results were produced using an NVIDIA Digits DevBox containing four Titan X GPUs with 12GB of memory per GPU, 64 GB DDR4 RAM, and a Core i7-5930K 3.5 GHz processor. Notice, that the best results in Table 2 were achieved with the FCN-8 VGG-16 network, which produced an average accuracy of 86.77%. The training time for these networks ranged from 7-10 days using a single Titan X for each network.

Table 2. Results of initial comparison.

Network type	Language/platform	Accuracy
FCN-VGG16	TensorFlow	86.77
D8 Frontend	TensorFlow	86.04
D8 Context	TensorFlow	81.50
CRF-RNN	Caffe	76.18

With respect to the FCN, both VGG-16 and VGG-19 networks were considered early in testing, but the VGG-16 quickly outperformed the VGG-19 network and was used for the remainder of the tests (as well as the base for the D8 and CRF-RNN networks). Additionally, these networks were trained using 8-bit Landsat 5/7 data. In an attempt to produce a solution that works for 16-bit Landsat 8 data, Top of Atmosphere (ToA) (Flood 2014) values were used to train a new FCN network. Here, the idea was to convert 8-bit data to ToA values for training so that trained network could classify 16-bit Landsat 8 values that are also converted to ToA. While this approach did not work on Landsat 8 data, it did improve the overall average accuracy of this network to 88.00%. Consequently, the remainder of the results and analysis on the FCN solution is based on the network trained with ToA values. Lastly, a sample of the results of this network from the validation set (and the associated GeoManitoba ground-truth labels) is given in Figs. 9 & 10, the full classification for the southern extent of Manitoba is given in Figs. 11 & 12, and the classification time (based on a Titan X) for the entire southern extent of Manitoba is 8 minutes and 42 seconds. Observe, that Fig. 11 was produced by tiles from the both the training and validation sets, while none of the tiles used to produce Fig. 12 were used in the training process.

The worst performing network in our experiments is the CRF-RNN, ironically this network was designed to extend FCNs for better classification accuracy by considering relationship between pixels (spatial cues). CRF-RNN in our case did not perform as expected. A possible explanation to this degradation in performance is the spectral variability inherent in the data. When considering low-spatial resolution (10-30 m) images associated with Landsat 5/7 satellite sensors the many spectral bands which gives variability to Land-cover classes significantly influences classification accuracy. The FCNs perform better in this case because the max pooling layers de-emphasize the importance of neighbouring pixels when considering fine details, thus placing more importance on spectral information. In contrast the low spatial resolution and multiple spectral scenes does not give to the strengths of CRF-RNN, hence the degradation in performance. From reviews of other works, CRF-RNN or CRF post-processing techniques are typically applied to high-spatial resolution (0.41 - 4 m) remote sensing imagery containing 3-4 spectral bands (see, *e.g.* Fu et al. (2017); Zhao et al. (2016, 2018)).

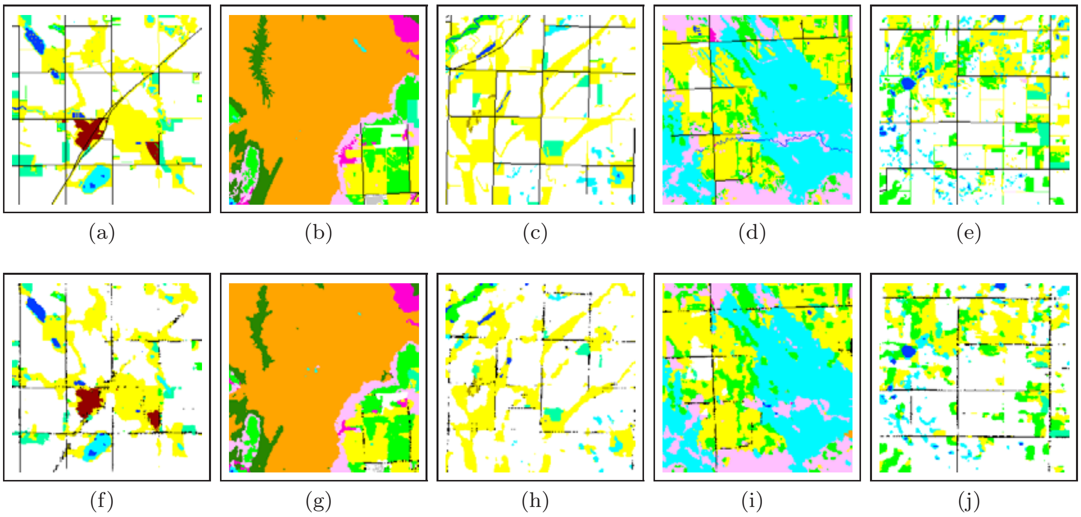
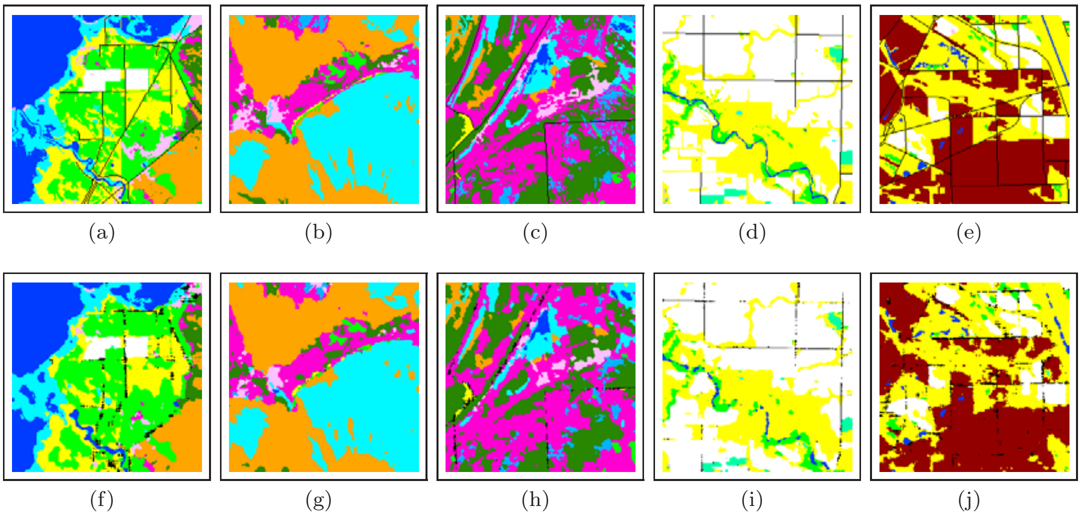


Figure 9. Sample validation set results. (Top row) semi automated labellings, and (bottom row) result using FCN network with ToA values.

Some interesting observations can be made about these results. First, there are two reasons for the loss of accuracy. One is the fact that FCN is not sensitive to fine details that exist at the single pixel level. This is due to the down-sampling and corresponding up-sampling present in the FCN network (Yu and Koltun 2015). This problem combined with the fact that the spatial resolution of the data is 30m means many of the roads and rivers – features that exist as a single pixel width – disappear as a result of the fine detail loss. Examples of this can be seen in Figs. 7 & 10. The second reason for the low accuracy is that there may be cases when the FCN output is correct and the original LULC maps is incorrect. GeoManitoba estimates their dataset is 90% accurate. Here, it should be also be noted, the fine details associated with roads and small bodies of water can be easily be added in post-processing through the application of image masks. Our experience is this process can improve the average accuracy by



(a) (b) (c) (d) (e)

(f) (g) (h) (i) (j)

Figure 10. Sample validation set results. (Top row) semi automated labellings, and (bottom row) result using FCN network with ToA values.

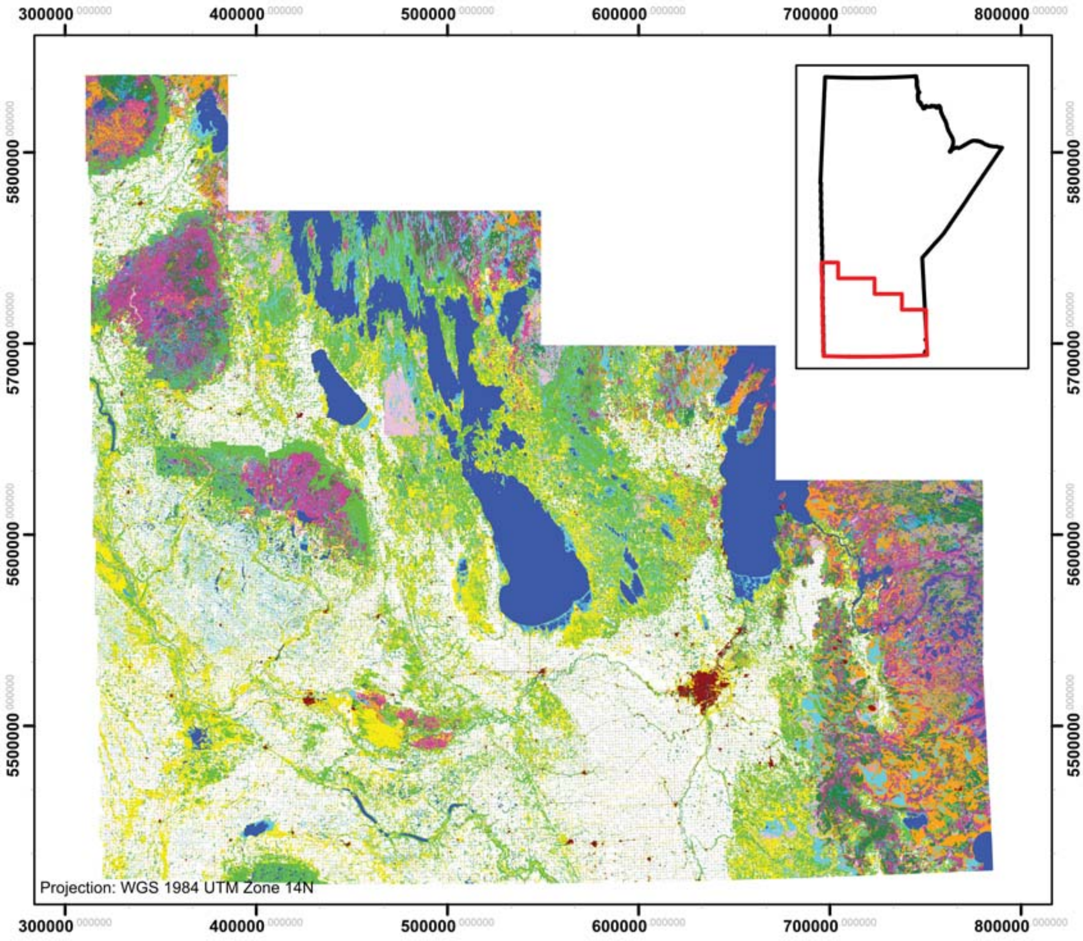


Figure 11. Final 2004 Landsat 7 LULC map produced by the FCN network using ToA input using both validation and training sets.

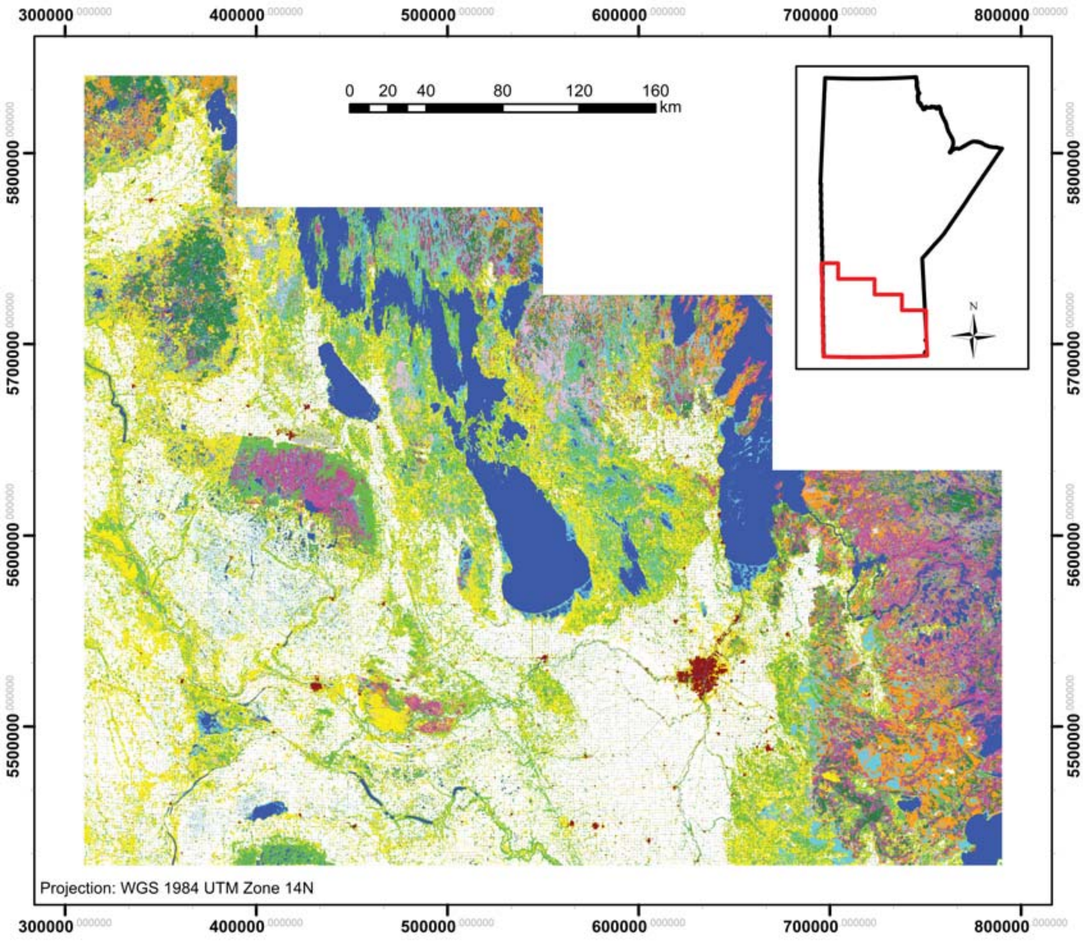


Figure 12. Final 2010 Landsat 7 LULC map produced by the FCN network trained on 2004 data using ToA input.

0.25%, and an example of this technique is given in Fig. 13.

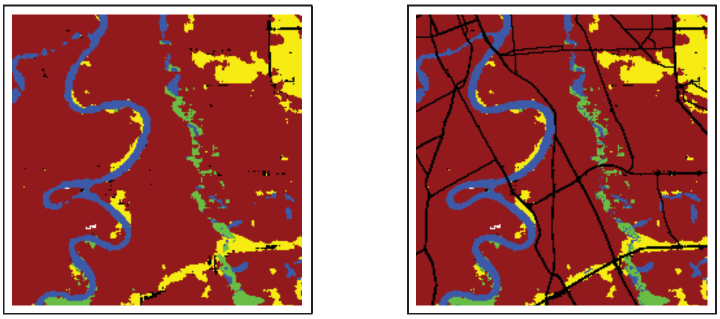

(a)
(b)

Figure 13. Example demonstrating post-processing of road and water classes.

Next, Tables 3 - 6 give the the percent accuracy for each class, the total number of pixels classified for each class, and the total number of pixels for each class in the 2004 LULC map provided by GeoManitoba. Note, for the first two tables, the rows correspond to ground truth and the columns to the FCN prediction. The values from the first two files were derived from the 2004 validation set, which contains 958 six banded satellite images of size 224x224. Here it is important to note a few things. First, the No data class is important from an overall system point of view since it is necessary for the network to be able to classify the pixels around the edge of a Landsat scene correctly when working with real data. However, this class raises the overall average of our approach since it is the easiest label for the network to learn to classify. Without this class, the average classification error drops to 87.35%. Similarly, the next two lowest classes are Burns and Road, with 44.34% and 57.20% accuracy, respectively. Notice, that Burns accounts for only $106/48,068,608 = 2.2 \times 10^{-4}\%$ of the total labels in the validation set. With respect to roads, this problem was discussed above (*i.e.* they are two narrow for the FCN-8 network to reliably detect) Removing, the No data, Burns, and Roads categories from the results produces an overall average accuracy of 87.96%. Observe that the above considerations have not dropped the overall average accuracy below 87%. Another important point regarding the class Burns is that this category represent only $390/153,158,638 = 2.5 \times 10^{-4}\%$ of the total pixels in the 2004 LULC map (*i.e.* ground truth labels) provided by GeoManitoba, which is quite close to the representation in the validation set. Thus, this class is significantly underrepresented in the original dataset, making it difficult to classify by the neural network.

Table 3. Percentage accuracy for each class from FCN network.

	No data	**Agriculture**	**Deciduous**	**Water**	**Grass**	**Mixedwood**	**Marsh**	**Tbog**	**Trock**	**Conifer**	**Burns**	**Open deci.**	**Forage**	**Cultural**	**Cutovers**	**Gravel**	**Road**	**Fens**	**Cloud**
No data	**99.94**	0.02	0.00	0.01	0.00	0.01	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Agriculture	0.00	**96.09**	0.69	0.02	1.72	0.01	0.43	0.00	0.00	0.00	0.00	0.02	0.24	0.01	0.00	0.00	0.76	0.00	0.00
Deciduous	0.00	2.07	**85.88**	0.30	6.10	2.24	1.21	0.08	0.01	0.15	0.00	0.95	0.34	0.03	0.15	0.02	0.45	0.00	0.00
Water	0.00	0.24	0.77	**95.98**	0.69	0.24	1.75	0.05	0.05	0.08	0.00	0.08	0.02	0.02	0.00	0.01	0.02	0.00	0.00
Grass	0.00	7.22	7.12	0.27	**80.36**	0.40	1.56	0.01	0.00	0.07	0.00	0.45	0.95	0.08	0.02	0.03	1.45	0.00	0.01
Mixedwood	0.00	0.06	5.52	0.34	0.86	**83.01**	1.45	1.21	0.86	4.66	0.00	1.07	0.02	0.02	0.51	0.02	0.38	0.00	0.00
Marsh	0.00	5.71	4.51	2.21	6.00	2.47	**75.59**	1.10	0.11	0.54	0.00	1.13	0.28	0.00	0.05	0.00	0.28	0.01	0.00
Tbog	0.00	0.00	0.44	0.12	0.03	2.32	1.63	**90.10**	0.69	3.74	0.00	0.70	0.00	0.00	0.14	0.00	0.06	0.00	0.00
Trock	0.00	0.00	0.34	0.74	0.05	10.28	0.47	3.82	**72.84**	10.81	0.00	0.10	0.00	0.00	0.40	0.00	0.13	0.00	0.00
Conifer	0.00	0.01	0.65	0.23	0.28	10.35	0.61	3.47	2.59	**80.35**	0.00	0.72	0.00	0.00	0.54	0.01	0.18	0.00	0.00
Burns	0.00	16.98	0.94	0.00	24.53	2.83	1.89	0.94	0.00	0.94	**44.34**	0.00	0.00	0.00	0.00	0.00	6.60	0.00	0.00
Open deci.	0.00	0.90	8.96	0.24	4.81	3.76	2.10	1.02	0.02	1.40	0.00	**75.93**	0.15	0.00	0.21	0.04	0.44	0.00	0.01
Foreage	0.00	4.21	1.79	0.04	5.62	0.04	0.37	0.00	0.00	0.00	0.00	0.08	**86.93**	0.02	0.01	0.00	0.88	0.00	0.00
Cultural	0.00	1.46	1.60	0.55	5.53	0.45	0.11	0.10	0.02	0.06	0.00	0.03	0.13	**86.93**	0.00	0.03	2.98	0.01	0.03
Cutovers	0.00	0.01	2.46	0.05	0.32	4.74	0.31	0.70	0.48	2.89	0.00	0.73	0.07	0.00	**86.86**	0.02	0.36	0.00	0.00
Gravel	0.00	1.93	5.44	3.63	10.05	1.44	0.41	0.13	0.03	0.65	0.00	1.49	0.06	0.39	0.36	**73.06**	0.92	0.00	0.01
Road	0.01	18.84	4.60	0.13	12.99	1.71	0.69	0.12	0.07	0.45	0.00	0.41	1.54	1.06	0.15	0.04	**57.20**	0.00	0.01
Fens	0.00	0.00	0.01	0.06	0.00	0.03	3.15	0.59	0.00	0.00	0.00	0.00	0.01	0.00	0.06	0.00	0.07	**96.02**	0.00
Cloud	0.02	2.01	4.49	0.12	8.83	0.20	1.55	0.00	0.00	0.01	0.00	0.37	0.56	0.12	0.00	0.11	0.59	0.00	**81.02**

Also, note several of the categories where misclassified to other similar categories. For example, the network misclassified the following similar classes.

- 4.66% of the Mixedwood class to the class Conifer
- 10.28% of the Conifer class to the class Mixedwood
- 7.22% of the Grass class to the class Agriculture
- 5.62% of the Forage class to the class Grass
- 8.96% of the Open deciduous class to the class Deciduous

These observations also underscore the importance of ground truth validation of the system output to result actual field data (rather than just using the labels provided by GeoManitoba).

Next, observe that the process of augmenting the dataset described in Section 6 did not result in overfitting. This was prevented by the use of an independent validation

Table 4. Total number of pixels classified for each class from FCN network.

	No data	**Agriculture**	**Deciduous**	**Water**	**Grass**	**Mixedwood**	**Marsh**	**Tbog**	**Trock**
No data	**1759000**	274	51	141	73	96	37	18	80
Agriculture	68	**14065000**	100930	3230	252420	1393	62349	15	5
Deciduous	273	137920	**5712100**	20150	405700	149220	80625	5203	710
Water	56	11229	36366	**4525100**	32452	11247	82478	2133	2446
Grass	207	500520	493470	18873	**5570500**	27681	107930	496	299
Mixedwood	82	1843	157030	9567	24566	**2359700**	41269	34489	24453
Marsh	64	166000	130960	64198	174220	71877	**2195900**	31896	3230
Tbog	63	52	6432	1796	496	33737	23668	**1308100**	10035
Trock	2	19	1288	2847	177	39404	1815	14627	**279150**
Conifer	16	207	10231	3698	4486	163140	9563	54649	40738
Burns	0	18	1	0	26	3	2	1	0
Open deci.	46	10995	109950	2918	59064	46191	25825	12507	270
Foreage	7	61092	26021	621	81651	525	5386	2	6
Cultural	0	3265	3576	1232	12392	1001	242	224	36
Cutovers	0	47	8364	184	1077	16128	1048	2393	1635
Gravel	0	572	1614	1077	2979	426	123	39	9
Road	87	172180	42006	1159	118720	15617	6332	1110	654
Fens	0	0	1	4	0	2	219	41	0
Cloud	4	405	904	24	1778	41	312	1	0

Table 5. Total number of pixels classified for each class from FCN network.

	Conifer	**Burns**	**Open deci.**	**Forage**	**Cultural**	**Cutovers**	**Gravel**	**Road**	**Fens**	**Cloud**
No data	113	0	6	2	0	10	0	85	0	1
Agriculture	116	14	3477	34855	1151	59	178	111740	0	188
Deciduous	10284	4	63272	22325	1670	9993	1590	30022	1	205
Water	3591	0	3925	913	939	199	598	1062	0	11
Grass	5007	22	31162	65791	5772	1063	1961	100330	0	692
Mixedwood	132450	2	30425	658	534	14386	461	10813	4	11
Marsh	15709	2	32680	8171	89	1405	94	8088	209	116
Tbog	54278	1	10194	23	11	1966	28	902	38	1
Trock	41435	0	365	11	14	1543	9	513	1	0
Conifer	**1266200**	0	11390	56	75	8574	111	2778	0	2
Burns	1	**47**	0	0	0	0	0	7	0	0
Open deci.	17137	0	**931660**	1878	61	2543	464	5389	1	73
Foreage	49	0	1134	**1262600**	276	206	32	12755	1	66
Cultural	134	0	63	297	**194870**	8	75	6672	14	61
Cutovers	9822	0	2498	226	9	**295690**	75	1210	0	6
Gravel	193	0	441	19	115	106	**21663**	272	1	2
Road	4129	2	3751	14044	9665	1333	325	**522780**	2	63
Fens	0	0	0	1	0	4	0	5	**6678**	0
Cloud	3	0	74	112	24	0	22	119	0	**16321**

set. Also, the lack of overfitting is further evidenced by the fact the network is able to successfully classify the 2010 Landsat 7 dataset, which was not used in the training process since there are no corresponding labels for this year. Taking overlapping tiles from 2004 increases the amount of data available for training, which is the fundamental reason for the improvements in accuracy. Deep neural networks require significant amounts of data to achieve high accuracy rates. As an example, approximately 1.2 million images were required before deep neural networks could perform well on the ImageNet problem, where the goal is to classify an unknown image into one of a thousand categories. In addition, overlapping tiles are possible for the GeoManitoba dataset due to the unique nature of producing LULC maps. From a pixel classification point of view, any satellite image of Manitoba is a valid image that requires classification. Thus, overlapping is not an issue since the new window/tile is also a valid problem. Here, the network learns to classify the pixels based on individual pixel feature values, the values of surrounding neighbours, low-level image characteristics (such as 6-banded "colours", edges, and texture), and high-level perceptual content in the image. Thus, overlapping tiles was a great way to augment the dataset (which

Table 6. Results of initial comparison.

	Total labels	Percentage of total
No data	67328000	30.54
Agriculture	46492000	21.09
Deciduous	21248000	9.64
Water	15454000	7.01
Grass	23441000	10.63
Mixedwood	9610600	4.36
Marsh	9515700	4.32
Tbog	5594000	2.54
Trock	1554400	0.70
Conifer	5910500	2.68
Burns	390	1.77×10^{-4}
Open deci.	4610900	2.09
Foreage	4819400	2.19
Cultural	694080	0.31
Cutovers	1020800	0.46
Gravel	122860	0.06
Road	2982000	1.35
Fens	36163	0.02
Cloud	51590	0.02

was required for higher accuracy), without overfitting the data.

Finally, the northwest corner of the the maps in Figs. 11 & 12 are significantly different. Since the results of the FCN have been validated this discrepancy is explained as follows. The majority of land cover change within the northwest section of the study area can be attributed to one of the following reasons. There is considerable land cover that was original classified as tree rock or mixedwood forest that are now classified as conifer forest. Over the period of 6 years between the 2004 and 2010 LULC maps, the trees may have matured sufficiently to become spectrally dominant. Secondly, there are a variety of smaller changes (treed bog to forest cover) than can occur naturally over time as landscape age and mature.

8. Conclusions

This article presented an approach for automating the production of LULC maps using an FCN. The best solution produced an average accuracy of 88%. Additionally, once trained, this approach can produce a map of the southern extent of Manitoba in 8 minutes and 42 seconds, which represents a phenomenal reduction in the 4,800 hours required by the current semi-automated approach. An important observation that the solution work presented here should be viewed as a solution to freeing up people from the tedious task of producing LULC maps, rather than eliminating a job. This solution will allow technicians to focus on analysis of problems and results rather than performing repetitive pattern classification, tasks which people find tedious and are prone to error. Future work will include extending the FCN training process to multiple GPUs to reduce the amount of training required. Additionally, future work will consist of performing real-world ground-truthing on the results to statistically evaluate the actual accuracy of the results. It could be the case that our system is performing better than 88% since GeoManitoba estimate their labels are approximately 90% accurate. Finally, producing a valid solution for 16-bit Landsat 8 is of utmost importance. The approach here was attempted on Landsat 8 data that was labelled ourselves, but the best accuracy that could be produced was 82.3%. This lower accuracy is likely due to the fact that the Landsat 8 labels were created by ourselves – rather than GeoManitoba

– and the fact that we normalized the 16-bit data to 8 bits. The immediate next steps of this work is to training a network to work directly on 16-bit data and to produce a dataset to train it.

Acknowledgements

This work was funded by a research contract with GeoManitoba, which is a department in the Manitoba provincial government. This work was dependent on the generous support and cooperation of GeoManitoba, including contributions by Greg Carlson, Jason Hancheruk, and Gerry Lux. Similarly, the work would not have been possible without help and support of the University of Winnipeg Research Office and Dr. Jino Distasio, as well as the in-kind contribution to the GeoManitoba research contract by the University of Winnipeg Research Office. Finally, Muthu Palaniappan and Mallikarjun Swamy participated in this work through the Mitacs Globalink Internship program.

References

Arnab, Anurag, Sadeep Jayasumana, Shuai Zheng, and Philip H. S. Torr. 2015. "Higher Order Potentials in End-to-End Trainable Conditional Random Fields." *CoRR* abs/1511.08119. http://arxiv.org/abs/1511.08119.

Bahdanau, Dzmitry, Jan Chorowski, Dmitriy Serdyuk, Philemon Brakel, and Yoshua Bengio. 2015. "End-to-End Attention-based Large Vocabulary Speech Recognition." *CoRR* abs/1508.04395. http://arxiv.org/abs/1508.04395.

Ball, John E, Derek T Anderson, and Chee Seng Chan. 2017. "Comprehensive survey of deep learning in remote sensing: theories, tools, and challenges for the community." *Journal of Applied Remote Sensing* 11 (4): 042609.

Ban, Yifang, Peng Gong, and Chandra Giri. 2015. "Global land cover mapping using Earth observation satellite data: Recent progresses and challenges." *ISPRS Journal of Photogrammetry and Remote Sensing* 103: 1 – 6. Global Land Cover Mapping and Monitoring, http://www.sciencedirect.com/science/article/pii/S0924271615000131.

Basu, Saikat, Sangram Ganguly, Supratik Mukhopadhyay, Robert DiBiano, Manohar Karki, and Ramakrishna Nemani. 2015. "DeepSat." In *Proceedings of the 23rd SIGSPATIAL International Conference on Advances in Geographic Information Systems - GIS '15*, 1–10. http://dl.acm.org/citation.cfm?doid=2820783.2820816.

Carreira-Perpiñán, M a, and G E Hinton. 2005. "On Contrastive Divergence Learning." *Artificial Intelligence and Statistics* 0: 17. http://learning.cs.toronto.edu/{~}hinton/absps/cdmiguel.pdf.

Castelluccio, Marco, Giovanni Poggi, Carlo Sansone, and Luisa Verdoliva. 2015. "Land Use Classification in Remote Sensing Images by Convolutional Neural Networks." *arXiv preprint arXiv:1508.00092* 1–11. http://arxiv.org/abs/1508.00092.

Chetlur, Sharan, Cliff Woolley, Philippe Vandermersch, Jonathan Cohen, John Tran, Bryan Catanzaro, and Evan Shelhamer. 2014. "cuDNN: Efficient Primitives for Deep Learning." *arXiv preprint arXiv: ...* 1–9. http://arxiv.org/abs/1410.0759.

Cho, Kyunghyun, Bart van Merrienboer, Çaglar Gülçehre, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. "Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation." *CoRR* abs/1406.1078. http://arxiv.org/abs/1406.1078.

Dai, Jifeng, Kaiming He, and Jian Sun. 2015. "BoxSup: Exploiting bounding boxes to supervise convolutional networks for semantic segmentation." In *Proceedings of the IEEE International Conference on Computer Vision*, Vol. 2015 International Conference on Computer Vision, ICCV 2015, 1635–1643.

Domingos, P. 2015. *The Master Algorithm: How the Quest for the Ultimate Learning Machine Will Remake Our World.* Basic Books. Basic Books. https://books.google.ca/books?id=glUtrgEACAAJ.

Dumoulin, Vincent, and Francesco Visin. 2016. "A guide to convolution arithmetic for deep learning." *ArXiv e-prints* .

Flood, Neil. 2014. "Continuity of Reflectance Data between Landsat-7 ETM+ and Landsat-8 OLI, for Both Top-of-Atmosphere and Surface Reflectance: A Study in the Australian Landscape." *Remote Sensing* 6 (9): 7952–7970. http://www.mdpi.com/2072-4292/6/9/7952.

Fu, Gang, Changjun Liu, Rong Zhou, Tao Sun, and Qijian Zhang. 2017. "Classification for high resolution remote sensing imagery using a fully convolutional network." *Remote Sensing* 9 (5): 498.

Goodfellow, Ian, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning.* MIT Press. http://www.deeplearningbook.org.

Han, X., Y. Zhong, and L. Zhang. 2016. "Spatial-Spectral Classification Based on the Unsupervised Convolutional Sparse Auto-Encoder for Hyperspectral Remote Sensing Imagery." *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences* 25–31.

He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015a. "Deep Residual Learning for Image Recognition." *CoRR* abs/1512.03385. http://arxiv.org/abs/1512.03385.

He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015b. "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification." *CoRR* .

Hong, Seunghoon, Hyeonwoo Noh, and Bohyung Han. 2015. "Decoupled Deep Neural Network for Semi-supervised Semantic Segmentation." *CoRR* abs/1506.04924. http://arxiv.org/abs/1506.04924.

Kirk, D. B., and W. W. Hwu. 2017. *Programming Massively Parallel Processors: A Hands-on Approach.* 3rd ed. Waltham, Massachusetts: Morgan Kaufmann.

Krizhevsky, A., I. Sutskever, and G. E. Hinton. 2012. "ImageNet Classification with Deep Convolutional Neural Networks." In *Advances in Neural Information Processing Systems 25*, 1097–1105.

LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton. 2015. "Deep learning." *Nature* 521 (7553): 436–444.

LeCun, Yann, Léon Bottou, Yoshua Bengio, and Patrick Haffner. 1998. "Gradient-based learning applied to document recognition." *Proceedings of the IEEE* 86 (11): 2278–2323.

Levine, Sergey, Peter Pastor, Alex Krizhevsky, and Deirdre Quillen. 2016. "Learning Hand-Eye Coordination for Robotic Grasping with Deep Learning and Large-Scale Data Collection." *CoRR* abs/1603.02199. http://arxiv.org/abs/1603.02199.

Li, F., and A. Karpathy. 2015. "CS231n: Convolutional Neural Networks for Visual Recognition." Course Lecture Notes, Stanford University.

Long, Jonathan, Evan Shelhamer, and Trevor Darrell. 2015. "Fully Convolutional Networks for Semantic Segmentation ppt." *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* 3431–3440.

Marmanis, Dimitrios, Mihai Datcu, Thomas Esch, and Uwe Stilla. 2016. "Deep learning earth observation classification using ImageNet pretrained networks." *IEEE Geoscience and Remote Sensing Letters* 13 (1): 105–109.

Papandreou, George, Liang-Chieh Chen, Kevin P Murphy, and Alan L Yuille. 2015. "Weakly- and Semi-Supervised Learning of a Deep Convolutional Network for Semantic Image Segmentation." *Proceedings of the IEEE International Conference on Computer Vision* 1742–1750.

Pathak, Deepak, Philipp Krahenbuhl, and Trevor Darrell. 2015. "Constrained convolutional neural networks for weakly supervised segmentation." In *Proceedings of the IEEE International Conference on Computer Vision*, Vol. 2015 International Conference on Computer Vision, ICCV 2015, 1796–1804.

Penatti, O. A. B., K. Nogueira, and J. A. dos Santos. 2015. "Do deep features generalize from everyday objects to remote sensing and aerial scenes domains?" In *2015 IEEE Conference* *on Computer Vision and Pattern Recognition Workshops (CVPRW)*, June, 44–51.

Russakovsky, Olga, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, et al. 2014. "ImageNet Large Scale Visual Recognition Challenge." *CoRR* abs/1409.0575. http://arxiv.org/abs/1409.0575.

Shelhamer, E., J. Long, and T. Darrell. 2017. "Fully Convolutional Networks for Semantic Segmentation." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39 (4): 640–651.

Simonyan, Karen, and Andrew Zisserman. 2014. "Very Deep Convolutional Networks for Large-Scale Image Recognition." *CoRR* abs/1409.1556. http://arxiv.org/abs/1409.1556.

Storie, C. D., and C. J. Henry. 2018. "Deep Learning Neural Networks for Land Use Land Cover Mapping." In *Proceedings of the 38th IEEE International Geoscience and Remote Sensing Symposium*, 4 pp. *accepted*.

Szegedy, Christian, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott E. Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. 2014. "Going Deeper with Convolutions." *CoRR* abs/1409.4842. http://arxiv.org/abs/1409.4842.

Treitz, Paul, and John Rogan. 2004. "Remote sensing for mapping and monitoring land-cover and land-use changean introduction." *Progress in Planning* 61 (4): 269 – 279. http://www.sciencedirect.com/science/article/pii/S0305900603000643.

Wang, Haohan, Bhiksha Raj, and Eric P. Xing. 2017. "On the Origin of Deep Learning." *CoRR* abs/1702.07800. http://arxiv.org/abs/1702.07800.

Yang, Yi, and Shawn Newsam. 2010. "Bag-of-visual-words and Spatial Extensions for Land-use Classification." In *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems*, GIS '10, New York, NY, USA, 270–279. ACM. http://doi.acm.org/10.1145/1869790.1869829.

Yosinski, Jason, Jeff Clune, Yoshua Bengio, and Hod Lipson. 2014a. "How transferable are features in deep neural networks?" In *Advances in neural information processing systems*, 3320–3328.

Yosinski, Jason, Jeff Clune, Yoshua Bengio, and Hod Lipson. 2014b. "How transferable are features in deep neural networks?" *CoRR* abs/1411.1792. http://arxiv.org/abs/1411.1792.

Yu, Fisher, and Vladlen Koltun. 2015. "Multi-Scale Context Aggregation by Dilated Convolutions." *CoRR* abs/1511.07122. http://arxiv.org/abs/1511.07122.

Zhang, L., L. Zhang, and V. Kumar. 2016. "Deep learning for Remote Sensing Data." *IEEE Geoscience and Remote Sensing Magazine* 4 (2): 22–40.

Zhao, J., Y. Zhong, H. Shu, and L. Zhang. 2016. "High-Resolution Image Classification Integrating Spectral-Spatial-Location Cues by Conditional Random Fields." *IEEE Transactions on Image Processing* 25 (9): 4033–4045.

Zhao, Ji, Yanfei Zhong, Tianyi Jia, Xinyu Wang, Yao Xu, Hong Shu, and Liangpei Zhang. 2018. "Spectral-spatial classification of hyperspectral imagery with cooperative game." *ISPRS Journal of Photogrammetry and Remote Sensing* 135: 31 – 42. http://www.sciencedirect.com/science/article/pii/S0924271616305767.

Zheng, Shuai, Sadeep Jayasumana, Bernardino Romera-Paredes, Vibhav Vineet, Zhizhong Su, Dalong Du, Chang Huang, and Philip H. S. Torr. 2015. "Conditional Random Fields as Recurrent Neural Networks." *CoRR* abs/1502.03240. http://arxiv.org/abs/1502.03240.

Zhu, Xiao Xiang, Devis Tuia, Lichao Mou, Gui Song Xia, Liangpei Zhang, Feng Xu, and Friedrich Fraundorfer. 2017. "Deep Learning in Remote Sensing: A Comprehensive Review and List of Resources." *IEEE Geoscience and Remote Sensing Magazine* 5 (4): 8–36.