Exploration of convolutional neural network architectures for large region map automation

Rostyslav-Mykola Tsenov, Christopher J. Henry, Joni L. Storie, Christopher D. Storie, Brent Murray, and Mikhail Sokolov

Abstract. Deep learning semantic segmentation algorithms have provided improved frameworks for the automated production of land use and land cover (LULC) maps, which significantly increases the frequency of map generation as well as consistency of production quality. In this research, a total of 28 different model variations were examined to improve the accuracy of LULC maps. The experiments were carried out using Landsat 5/7 or Landsat 8 satellite images with the North American land change monitoring system (NALCMS) labels. The performance of various convolutional neural networks and extension combinations were assessed, where Visual Geometry Group Network with an output stride of 4, and modified U-Net architecture, provided the best results. Additional expanded analysis of the generated LULC maps was also provided. Using a deep neural network, this work achieved 92.4% accuracy for 13 LULC classes within southern Manitoba representing a 15.8% improvement over published results for the NALCMS. Based on the large regions of interest, higher radiometric resolution of Landsat 8 data resulted in better overall accuracies (88.04%) compared to Landsat 5/7 (80.66%) for 16 LULC classes. This represents an 11.44% and 4.06% increase in overall accuracy compared to previously published NALCMS results, including larger land area and higher number of LULC classes incorporated into the models compared to other published LULC map automation methods.

1 Introduction

Land use and land cover (LULC) maps are often the output products generated from raw satellite imagery where, in this case, each pixel of the image is assigned to a land cover class or label (e.g., water, forest, road, cropland, etc.). LULC map information is crucial for many applications such as natural resource management, wildlife habitat protection, urban expansion, hazard and damage delineation, etc.1 For many years, LULC maps were generated manually using semi-automated techniques such as supervised, unsupervised and hybrid algorithms (e.g., K-means clustering, decision tree and maximum likelihood classifier).2 However, these methods take a lot of time and effort to produce adequate accuracies but are rarely consistent over different production years or producers. Therefore, map automation is on the rise as it significantly increases the production frequency and consistency of LULC products. Methods of map automation include using advanced algorithms, machine learning tools,2 or deep learning (DL)3 started growing in terms of development and application in the last decade. DL has achieved success in the field of remote sensing by providing fast and consistent performance and solving problems, such as image fusion, image registration, scene classification and object detection, LULC classification, etc.3 Research on DL has resulted in groundbreaking papers published every few months in the fields of image processing,4 computer vision,5 and object detection,6 which has led to the development of DL applications in remote sensing. Most issues in DL map applications are

*Address all correspondence to Christopher J. Henry, ch.henry@uwinnipeg.ca

© 2023 Society of Photo-Optical Instrumentation Engineers (SPIE) [DOI: 10.1117/1.JRS.17.018505]
solved using convolutional neural networks (CNNs) as CNNs are considered the state-of-the-art method for computer vision tasks, such as segmentation, image classification, and object recognition\(^7\).

To resolve the semantic segmentation task of LULC mapping, Long et al.\(^8\) and Badrinarayanan et al.\(^9\) introduced an encoder-decoder architecture, which extracts features from the imagery using CNN in the form of an encoder and then decodes those features back to the original size using a reversed CNN. The implementation of encoder-decoder networks in remote sensing\(^10\)–\(^14\) has achieved great success with accurate and consistent results between 80% and 90% for LULC mapping. These successes have led to the implementation of DL models for mapping of LULC within commercial software, for example, ArcPro (Esri). However, in comparison to the published literature discussed above, commercially available DL models have lower overall classification accuracies, typically ranging from 78% to 88%\(^15\) and have even lower performance with a larger number of LULC classes.

The goal of this work was to develop and improve the results of land cover mapping using DL techniques based on label or training data from the North American land change monitoring system (NALCMS).\(^16\) The NALCMS labels were chosen for training because of the large number of potential land cover categories (20 classes), ranging from arctic to tropical ecosystems, and because the NALCMS provides a much larger land area extent than other labelled/training datasets. The 30-m NALCMS maps are currently only generated every 5 years, by the three-country consortium, due to cloud cover, technical and resource constraints.\(^17\) Other users are motivated to increase the frequency of production of the NALCMS products at accuracies that can only be obtained by developing automated DL approaches. For example, the overall classification accuracy of 82% has been achieved for Mexico using MODIS data\(^18\) and 76.6% accuracy for all of North America\(^17\) using Landsat\(^17\) both using 30-m NALCMS.

To achieve the goal of improving overall accuracy (OA), this work incorporated new and modified CNNs\(^19\)–\(^22\) to the existing architecture, and tested a variety of more advanced state-of-the-art model extensions and their combinations.\(^23,24\) We experimented with CNNs on two different datasets, one that uses 8-bit data from Landsat 5/7, and the second uses 16-bit data from Landsat 8, both with a spatial resolution of 30 \(\times\) 30 m. Data from these two sensors have similar spectral resolutions with bands in the red, green, blue and near- and shortwave-infrared regions. After determining the best model variation on the smaller datasets, two models were trained on the larger Landsat 5/7 Prairie dataset and the Landsat 8 Lake Winnipeg dataset. Landsat 5/7 Prairie dataset model resulted in a 80.66% OA, which is equivalent to the performance of ArcGIS/Pro deep learning package\(^15\), while Landsat 8 Lake Winnipeg dataset model performance reached 88.04% OA. The Landsat 8 data increased the OA by 11.44% compared to\(^17\) and are 6% to 8% higher than its Landsat 5/7 counterpart. Producing models for both Landsat 5/7 and Landsat 8 data means that historical and current LULC maps can be generated back to 1984 with more frequency, consistency and accuracy than ever before.

### 2 Use of DL in Semantic Segmentation and Remote Sensing

#### 2.1 Deep Learning

In the last two decades, the use of NNs and DL has grown due to a large number of advances in many research areas\(^25\) such as large datasets, improvements in the graphics processing unit (GPU) technology, and models, which previously took weeks or months to train, now take just a few hours or days. In 2012, the AlexNet\(^26\) model was presented and won the ImageNet large scale visual recognition challenge. It was considered a revolutionary model that first showed NNs could outperform other algorithms in image recognition and classification tasks. This new architecture was only able to be trained due to the advent of general purpose computing using GPU. Similar advances in CNN architecture occurred in 2014 when the Visual Geometry Group Network (VGGNet) model\(^27\) and GoogleNet (also known as the Inception model\(^28\)) where introduced. Recently, Francois Chollet introduced an extension of the Inception architecture called Xception\(^21\) and He et al.\(^29\) introduced groundbreaking architecture that resembles VGGNet, but is approximately eight times deeper.
CNNs were first successful at the recognition of handwritten digits using LeNet developed by LeCun in 1998\textsuperscript{30} using CPUs. DL using GPUs advanced applications in speech recognition,\textsuperscript{31} natural language processing,\textsuperscript{32} image processing,\textsuperscript{4} computer vision,\textsuperscript{5} and object detection.\textsuperscript{6} As a result, remote sensing applications were a natural extension of image and object recognition\textsuperscript{3,12–14} especially in fields like image preprocessing and classification. This work focuses on mapping of LULC within the branch of classification in DL for remote sensing.

2.2 Deep Convolutional Encoder-Decoders

Many computer vision tasks cannot be solved by conventional CNNs, such as object detection or semantic segmentation. However, with architectural structural changes to the networks, they can be used in these domains. To solve these tasks, the deep convolutional encoder-decoder architecture was introduced.\textsuperscript{8} The encoder-decoder architecture is used to extract features from the input using a CNN and then decode those features back to the original size using a modified CNN before classification. This concept was later improved\textsuperscript{8,19} by slightly altering existing architecture through the introduction of more layers to the decoder. At the same time, Deeplab\textsuperscript{24,33} focused more on modifying the encoder with the CNNs such as ResNet,\textsuperscript{20} and Xception.\textsuperscript{21} Similarly, DeepLab expanded the architecture with network extensions, such as atrous spatial pyramid pooling (ASPP) and output stride (OS) to capture objects and context of the features at multiple scales. Concurrently, researchers starting focusing on areas other than end-to-end network architectures. For instance, some researchers focused on network extensions, which are plug-in modules that aim to improve the performance of the deep convolutional encoder-decoders without significant changes to the existing architecture. In 2015, the context aggregation extension module\textsuperscript{34} was presented to increase the sharpness of predictions by aggregating multiscale contextual information. Additionally, Goodfellow et al.\textsuperscript{35} introduced a generative adversarial network (GAN).\textsuperscript{36} A GAN consists of a generator and a discriminator, where the goal of the generator is to produce an image as close to the real one as possible, and the discriminator attempts to distinguish between original and generated images. The progressive GAN\textsuperscript{23,37} improved the original GAN by gradually adding layers to both the generator and discriminator throughout training, which helps with training stability.

2.3 CNNs in Remote Sensing

The success of DL in image processing encouraged researchers to use it in remote sensing. The development of CNNs was mainly focused on scene classification and LULC map production. One of the first studies classified LULC using high-resolution satellite images\textsuperscript{10,11} due to good structural information. At that time, conventional DL algorithms performed poorly on medium-resolution satellite images (10 to 30 m per pixels) because of a deficiency of such structural information. In 2017, Sharma et al.\textsuperscript{38} presented a patch-based CNN, which performed effectively on medium-resolution satellite images. Also, the multisize/scale ResNet ensemble architecture was invented by Refs.\textsuperscript{39} and \textsuperscript{40} to perform a single-pixel prediction based on an input image cropped to a specific size.

Most of the modern architectures follow the encoder-decoder structure with some minor changes. Examples of one of the first to successfully use semantic segmentation architectures for remote sensing applications are given by Storie and Henry,\textsuperscript{12} Henry et al.,\textsuperscript{13} Alhassan,\textsuperscript{41} and Alhassan et al.\textsuperscript{14} Their solution is based on a fully convolutional network (FCN) structure with different encoders (i.e., VGGNet, GoogleNet, ResNet) used on Landsat 5/7 satellite images. Also, they implemented GAN and context module extensions to boost the performance of predictions. Additionally, work presented in 2018\textsuperscript{42} showed that architectures that were successful in semantic segmentation on the real-world images need to be modified to get their full potential in remote sensing. Many different modifications were made to improve the performance of the architectures on satellite imagery. For example, to keep most of the structural information, a no-downsampling encoder network that relies on atrous convolution was developed.\textsuperscript{43} Some works centred on modifying the decoder by adding symmetric unconvoluted layers.\textsuperscript{44,45} These architectures significantly improved the accuracy of generated LULC maps.
In the field of remote sensing, there are a number of datasets and state-of-the-art DL models that focus on resolving object detection and semantic segmentation problems. Table 1 compares the most popular datasets and DL algorithms from literature and software. One trend that can be seen is that the lower the number of classes, the better the performance of the models. When the number of LULC classes exceeds 10, only ResNet-FCN was shown to obtain an OA higher than 85%. Also, other models that provide accurate results follow the U-Net architecture,19 Fast Region-based Convolutional Network method (Fast R-CNN),52 and Mask R-CNN.53 The last two derive from R-CNN54 architecture that relies on splitting images into segments then processing each by a DCNN.

This work is a continuation of the research by Alhassan et al.,14 which was able to reach OA > 90% on the dataset with 19 classes. While these results are good, the work was conducted on a relatively small geographical area ("148,800 km²). In this paper, Landsat 5/7 data were used for mapping the geographic extent of both southern Manitoba (Fig. 1) and the Prairies (Fig. 2), while Landsat 8 was used for the entire Lake Winnipeg/Churchill-Nelson watershed (not shown).

### 3 Implementing and Training the Models

#### 3.1 Data

As the overall goal of this work was to develop models to automate LULC map production corresponding to the NALCMS labels; the models were developed using both the 2010 NALCMS labels with Landsat 5/7 data (2009-2011), as well as with 2015 NALCMS labels with Landsat 8 (2014-2016) data. In the Prairie (Fig. 2) and Lake Winnipeg/Churchill-Nelson watersheds, 15 of the 19 NALCMS classes were present and a cloud label was added to make it 16 classes. Cloud labels were generated using Otsu thresholding techniques on the blue band55 for the Landsat 5/7 dataset; cloud identification was done using the quality assessment (QA) band for the Landsat 8 dataset.

Due to the associated training time required for an area the size of the watershed dataset, to allow for direct comparison to,41 and to expand to larger regions of interest, we used three datasets in our analysis. The datasets were: (i) the southern Manitoba (148,800 km²) and (ii) the three prairie provinces of Alberta, Saskatchewan and Manitoba (~1,960,000 km²) based on 2010 Landsat 5/7, and (iii) the Lake Winnipeg/Churchill-Nelson watershed (most of the prairies,

### Table 1  Comparison of popular remote sensing datasets and their state-of-the-art DL models, where most of the datasets are used as benchmarks. This table includes metrics for comparison such as OA, mean IoU (mIoU), and F1-score.

| Dataset       | Model                | Classes | OA      | F1-score | mIoU | References |
|---------------|----------------------|---------|---------|----------|------|------------|
| INRIA         | ICT-Net             | 1       | 97.14   | —        | 80.32| 46         |
| AIRIS         | ICT-Net             | 1       | 95.7    | —        | 91.7 | 46         |
| LandCover.ai  | ResNet-50           | 4       | 85.56   | —        | —    | 47         |
| ISPRS Potsdam | ResUNet-a           | 5       | 91.5    | 92.9     | —    | 48         |
| ISPRS Vaihingen| TreeUNet            | 5       | 90.4    | 89.3     | —    | 49         |
| GID-C         | Dual attention Deep fusion | 5       | 85.49   | 85.37    | 74.45| 50         |
| DeepGlobe     | Deep aggregation net| 6       | —       | —        | 52.72| 51         |
| GID-F         | Dual attention deep fusion | 15     | 83.93   | 83.23    | 71.29| 50         |
| CLC 2018      | U-Net               | 15      | 82.41   | 77.27    | —    | 15         |
| NLCD 2016     | U-Net               | 16      | 78.1    | 67.69    | —    | 15         |
| GeoManitoba   | ResNet-FCN          | 19      | 90.46   | —        | 75.66| 41         |

In the field of remote sensing, there are a number of datasets and state-of-the-art DL models that focus on resolving object detection and semantic segmentation problems. Table 1 compares the most popular datasets and DL algorithms from literature and software. One trend that can be seen is that the lower the number of classes, the better the performance of the models. When the number of LULC classes exceeds 10, only ResNet-FCN was shown to obtain an OA higher than 85%. Also, other models that provide accurate results follow the U-Net architecture,19 Fast Region-based Convolutional Network method (Fast R-CNN),52 and Mask R-CNN.53 The last two derive from R-CNN54 architecture that relies on splitting images into segments then processing each by a DCNN.

This work is a continuation of the research by Alhassan et al.,14 which was able to reach OA > 90% on the dataset with 19 classes. While these results are good, the work was conducted on a relatively small geographical area (~148,800 km²). In this paper, Landsat 5/7 data were used for mapping the geographic extent of both southern Manitoba (Fig. 1) and the Prairies (Fig. 2), while Landsat 8 was used for the entire Lake Winnipeg/Churchill-Nelson watershed (not shown).
plus parts of eastern Ontario and North Dakota) using 2015 Landsat 8 (∼1,400,000 km²). Landsat 5/7 data were obtained for the southern Manitoba dataset (Fig. 1), which is the same data and area coverage used Alhassen et al., to see the impact of model-architecture experiments on results. Most of the experiments were performed on the Landsat 5/7 southern Manitoba dataset to reduce the training time required for the comparison of models and model extensions.
Then, based on the results, the best architecture was chosen, and the models for Landsat 5/7 (Prairies) and Landsat 8 (Lake Winnipeg watershed) were generated.

### 3.2 Data Preprocessing

All satellite data and corresponding labels were preprocessed before training. A popular input-size for encoder-decoder networks is $224 \times 224$ pixel rows and columns, so the first step was to tile the larger dataset extents. For example, the mosaic forming the southern Manitoba dataset has 14,975 rows $\times$ 13,331 columns of pixels. By creating tiles of $224 \times 224$ pixels, there is a suitable number of images to train a DL model, and images of this size can be processed by our computer hardware. Consequently, we used the data augmentation technique called tiling with some minor improvements.

#### 3.2.1 Tiling

Tiling is the process of splitting a larger image into smaller-sized, square images (Fig. 3), also referred to as tiles, where a tile is one $224 \times 224$ image (pixel row and column). The southern Manitoba dataset ($14,975 \times 13,331$ pixels) generated 3630 disjoint tiles with meaningful data. Moreover, increasing the number of tiles generated from the same region of interest extent can be used to improve feature detection. For example, tiling with 1/2 overlap (shifted tiling) was used to increase the size of the dataset by a factor of 4 (Fig. 3), and flipping of tiles increased it by another factor of 2. Using these processes, a dataset of 29,100 training tiles and 2100 validation tiles was generated. Additionally, the Landsat 5/7 Prairie dataset was generated using only regular tiling and consisted of 85,400 training tiles and 8,500 validation tiles, while the Landsat 8 Lake Winnipeg watershed dataset was generated using shifted tiling and consisted of 107,900 training tiles and 12,000 validation tiles. Splitting tiles for training and validation was performed using a stratified random process to ensure that training and validation sets contain all LULC classes and tiles from all locations. The splitting ratio was 90% training to 10% validation, which followed previous work. This splitting ratio allows effective training of the LULC classes with low presence or representation.

#### 3.2.2 Transformation

Training DL models often results in either overfitting or underfitting that is caused by a lack of variation in the dataset or inadequate model architecture. One of the simplest ways to resolve this issue is to shuffle data after every epoch and apply random data transformation on each image. Data transformation ensures that tiles are not represented in the same way during training by randomly applying rotation, flipping, or zoom cropping (Fig. 4). Note that, unlike augmentation (shifted tiling), transformation is used during training and does not increase the number of generated tiles.

### 3.3 Experimental Design

Training of the networks was performed using the TensorFlow Python library with Docker containers. Most of the results were generated using an NVIDIA Digits DevBox containing four

---

**Fig. 3** Illustration of the tiling process.

---
Titan X GPUs with 12 GB of memory per GPU, 64 GB DDR4 RAM, and a Core i7-5930K 3.5 GHz processor. The training time on the southern Manitoba datasets varied between 5 and 7 days, while the Prairie and Lake Winnipeg watershed dataset took between 20 and 30 days. The prediction time of the network on the non-augmented southern Manitoba dataset took 8 minutes, while the prediction of the Prairie or Lake Winnipeg watershed datasets took close to 1.5 hrs for both sensors. Most of the networks were trained with a batch size of 16 on a model that was distributed across 3 GPUs, leading to a global batch size of 48. However, some network variations used a lower batch size due to computational complexities. As per Alhassan, training was performed with a learning rate of $\eta = 10^{-4}$ for 100 epochs, then with a learning rate of $\eta = 10^{-5}$ for another 100 epochs using the Adam optimization algorithm.

4 Experiments, Results, and Analysis

The experiments evaluated and compared the performance of different model variations, which were trained on the Landsat 5/7 southern Manitoba. Then, based on the results, the best model combination was selected to train models using Landsat 5/7 Prairie and Landsat 8 Lake Winnipeg watershed datasets. The model trained on the Landsat 5/7 dataset will allow us to generate LULC maps from 1984 to 2017, and the model trained on the Landsat 8 dataset will allow us to generate LULC maps starting from 2013 and onwards. Lastly, the results of all generated products are discussed, assessed, and compared to existing state-of-the-art models. The overall training process is depicted in Fig. 5.

4.1 Evaluation Metrics

The experiment results were evaluated using OA, precision, recall, F1-score, and critical success index (CSI) [CSI is also known as a threat score, and identical to intersection over union (IoU)] were used to evaluate the results. The OA is a metric that calculates the ratio between the number of correctly classified pixels and a total number of pixels in the image, thus providing a general assessment of the prediction, which can be easily compared and understood. Define $n_{ij}$ as the number of pixels in the image with ground truth label $i$ and corresponding predicted label $j$. Let $t_i = \sum_{j=1}^{C} n_{ij}$ denote the total number of pixels labelled with label $i$, $C$ is the number of classes,
\( n_{ii} \) are the number of pixels correctly predicted, and \( n_{ji} \) are the number of incorrectly label pixels (with respect to label \( j \)).\(^{41}\) The OA is defined as

\[
\text{OA} = \frac{\sum_{i=1}^{C} n_{ii}}{\sum_{i=1}^{C} t_i}.
\] (1)

Using the OA, the best model combination was determined, which was later trained on larger datasets to save time for the experiments. Next, we will define, precision,\(^{61}\) recall,\(^{61}\) F1-score,\(^{62}\) and CSI,\(^{63}\) where the CSI score is intersection over union (IoU).\(^{64}\) Let class \( i \) be labelled a positive class, and every other class labeled a negative class. Then, the true positive, \( TP_i \), is defined as the number of correctly predicted pixels of the positive class, true negative \( TN_i \) as the number of correctly predicted pixels of the negative class, false positive \( FP_i \) as the number of incorrectly predicted pixels of the positive class, and false negative \( FN_i \) as the number of incorrectly predicted pixels of the negative class. Given these definitions, we can then define the following metrics:

\[
\text{Precision}_i = \frac{TP_i}{TP_i + FP_i},
\] (2a)

\[
\text{Recall}_i = \frac{TP_i}{TP_i + FN_i},
\] (2b)

\[
F1_i = 2 \cdot \frac{\text{Precision}_i \cdot \text{Recall}_i}{\text{Precision}_i + \text{Recall}_i},
\] (2c)

\[
\text{CSI}_i = \frac{TP_i}{TP_i + FP_i + FN_i}.
\] (2d)

Both the average and weighted average of each metric were calculated as
Average \[ = \frac{1}{l} \sum_{i=1}^{l} M_i, \]  

Weighted average \[ = \sum_{i=1}^{l} w_i \times M_i, \]  

where \( l \) is the number of classes, \( M_i \) is the metric value of the class \( i \), and \( w_i \) is the pixel proportion of the class \( i \). Note, all the metrics were calculated on a per tile basis, and to get global dataset evaluation, each class \( i \) was calculated using a weighted average.

4.2 Comparison of Model Variations

All model combinations were trained and validated on the southern Manitoba dataset. The experiment included the comparison of encoders, such as VGGNet, GoogleNet, ResNet, Xception, and decoders, such as FCN-8, U-Net, and feedbackward decoder. Table 2 shows the results of the trained decoders paired with the VGGNet encoder. As can be seen, the top two decoders (U-Net and modified U-Net) differ in accuracy by only 0.04%, which is negligible for machine learning applications. Moreover, our implementation of the U-Net model has approximately 43 million trainable parameters and 12 billion floating point operations (FLOPS) were performed during training (see Lu et al. for similar comparisons). In comparison, the modified U-Net has ~32 million trainable parameters and 9 billion FLOPS, making it 25% more efficient in terms of both storage and processing speed. Thus, with all else being equal, it is better to choose a network with fewer parameters based on storage and model capacity requirements as well as reduced training and inference time. Consequently, the modified U-Net decoder was chosen as the main decoder network because it has three times fewer parameters. Table 3 displays the results of all implemented encoders in combination with the modified U-Net decoder. VGGNet proved to have the highest OA of 90.11%, followed by ResNet at 89.05%.

VGGNet and ResNet CNNs with the modified U-Net decoder were chosen as backbone models for future experiments. In these experiments, both of these networks were trained with

| Table 2 | Results of trained VGGNet encoder in combination with different decoders on the Landsat 5/7 data from the southern Manitoba. |
|---------|---------------------------------------------------------------|
| Network | OA                |
| FCN-8     | 88.28             |
| U-Net     | 90.15             |
| Feedbackward | 89.59         |
| Modified U-Net | 90.11     |

| Table 3 | Results of trained modified U-Net decoder in combination with different encoders on the Landsat 5/7 data from the southern Manitoba. |
|---------|---------------------------------------------------------------|
| Network | OA                |
| VGGNet | 90.11             |
| GoogleNet | 83.14      |
| Xception | 88.36          |
| ResNet | 89.05             |
Additionally, networks were trained with different model and architectural extensions, such as ASPP, Context Module, GAN, progressive GAN, and DeepLabv3+ (Table 4). Based on the results, adversarial training had little or no effect at all on the models that performed well on the given dataset. Also, the ResNet Deeplabv3+ architecture was trained on our dataset and showed low performance, reaching 87.43% of OA. The model with the best performance and lowest complexity was VGGNet with modified U-Net and OS 4, which reached 92.4% on the southern extent of the Manitoba validation dataset. As the OA of the model approaches 100% it becomes more difficult to improve the performance by introducing changes and enhancement to the model and architecture. In our experiments, we can see that extensions, like ASPP, context module and GAN can improve the performance of the model. However, if the model already performs well, implementing or even combining them will not provide further benefit, and in some cases even aggravate the results. The best performing network architecture is depicted in Fig. 6. The best results generated by the model based on OA are shown in Fig. 7(a), where the model easily recognized edges of the water bodies, rivers and roads. Similarly, the worst results based on OA are shown on the right of Fig. 7(b), where the model did predict most of the area well but missed and misclassified some parts of the roads and water bodies.

### 4.3 Training Model on a Large Dataset

#### 4.3.1 Landsat 5/7

After determining that the VGGNet encoder with modified U-Net decoder and OS 4 was the best, this model combination was then used to train models on the Landsat 5/7 Prairie dataset. Table 5 (left) shows pixel proportion of each class, Table 6 (left) shows the per-class assessment of the Landsat 5/7 Prairie dataset and Table 7 shows a global assessment of the predicted map. Based on these results, LULC classes that performed well were found to have a higher percentage of pixels or representation than other LULC classes which had a small number of pixels. It has been documented that deep learning models need a large number of class examples (pixels) to achieve good performance for that class. Forest-related classes with low presence were the classes most often misclassified. Temperate or subpolar needleleaf forest class and the urban and built-up classes were misclassified with cropland. Moreover, prediction of the cloud class had extremely high accuracy because the clouds were discriminated based on labels generated using a thresholding method; the network did not have difficulties generating this cloud class due to it being a straightforward function with sufficient data examples. Overall, the performance of the

---

**Table 4** OA results of trained encoder-decoder models in combination with different extensions on the Landsat 5/7 data from the southern Manitoba.

| Network + extension       | OS 32 | OS 16 | OS 8  | OS 4  | OS 2  |
|---------------------------|-------|-------|-------|-------|-------|
| VGGNet27                  | 90.11 | 90.66 | 91.25 | 92.4  | 91.01 |
| VGGNet + ASPP24           | 90.72 |  —    |  —    | 92.22 |  —    |
| VGGNet + context module34 | 90.18 |  —    |  —    | 92.04 |  —    |
| VGGNet + GAN36            | 89.99 |  —    |  —    | 92.37 |  —    |
| VGGNet + progressive GAN33,37 | 90.09 |  —    |  —    | 92.41 |  —    |
| ResNet20                  | 89.05 | 89.72 | 90.02 | 91.42 |  —    |
| ResNet + ASPP24           |  —    |  —    |  —    | 91.37 |  —    |
| ResNet + context module34 |  —    |  —    |  —    | 90.31 |  —    |
| ResNet + GAN36            | 89.26 |  —    |  —    | 91.04 |  —    |
| ResNet + Deeplabv3+24     |  —    |  —    |  —    | 87.43 |  —    |
The model reached 80.66% OA on the validation dataset, where only a few tiles had no data pixels. No data are the background class that covers empty areas on the image. The accuracy presented in this dataset is lower due to the size and amount of extractable features presented in this dataset; particularly the discrimination of clouds. It was noted that the cloud labels generated by the Otsu thresholding method did not classify some of the clouds, especially transparent ones, and misclassified bright urban areas as clouds. This model showed the performance to be slightly better than published by\textsuperscript{17} but equivalent to the models present in ArcGIS/Pro deep learning package.\textsuperscript{15}

The best results generated by the model based on OA are shown in Fig. 8(a), where the model had no difficulty in predicting tiles with a low number of classes, and perfectly segmented water, forests and croplands. Similarly, the worst results based on OA are shown in Fig. 8(b), where the

Fig. 6 Network architecture diagram of the structure for the best model combination: VGGNet encoder with modified U-Net decoder and OS 4 (a), and description of each block (b).

Fig. 7 Examples of good (a) and poor (b) results generated by the best model trained on the Landsat 5/7 data from the southern Manitoba. The first row are ground truth tiles, and second row are predicted tiles.
4.3.2 Landsat 8

Next, the same VGGNet encoder with modified U-Net decoder and OS 4 model was trained on the Landsat 8 Lake Winnipeg watershed, which has higher radiometric resolutions (16-bit vs 8-bit of Landsat 5/7). The trend in the results provided by the model trained on the Landsat 8 dataset is similar to the Landsat 5/7 model and can be seen in Tables 6 and 7 (right). The OA of the predicted map increased by 8%, and improvements in accuracy were noticed across most of the LULC classes. Because the land cover classes remained consistent, we speculate that these improvements are achieved due to the increased radiometric variability provided by the 16-bit instead of the 8-bit dataset, which made it easier to distinguish features due to a larger range of values. There were a lot of studies that explored the impact of radiometric resolution on the model performance, where results indicated that higher radiometric resolution can be beneficial to certain tasks, such as texture-based classification, and have an almost negligible effect on most remote sensing problems. The research also implies that both spectral and spatial resolutions play a far more important role, but in our case, both of the sensors (Landsat 5/7 and Landsat 8) have similar spectral and spatial resolutions. Recall, the cloud detection for Landsat 8 was trained using data generated from the Landsat 8 QA band provided by the sensor and not the thresholding method. Despite the results being lower than Landsat 7, the performance on the cloud class is still one of the best for this model. This model showed the performance higher than results reported by (11.44% increase) and the models present in ArcGIS/Pro deep learning.
Table 6  Per-class assessment of the models trained on the Landsat 5/7 Prairie and Landsat 8 data from the Lake Winnipeg watershed.

| Name of the class                                      | Landsat 5/7       |          |          |          | Landsat 8       |          |          |          |
|--------------------------------------------------------|-------------------|----------|----------|----------|-----------------|----------|----------|----------|
|                                                        | CSI               | Precision| Recall   | F1       | CSI             | Precision| Recall   | F1       |
| No data                                                | 99.76             | 99.78    | 99.98    | 99.88    | 99.99           | 99.99    | 99.99    | 99.99    |
| Temperate or subpolar needleleaf forest                 | 74.99             | 84.15    | 87.33    | 85.71    | 76.21           | 84.65    | 88.44    | 86.5     |
| Subpolar taiga needleleaf forest                        | 42.78             | 62.8     | 57.29    | 59.92    | 41.62           | 67.21    | 52.22    | 58.77    |
| Temperate or subpolar broadleaf forest                  | 57.41             | 75.7     | 70.39    | 72.95    | 71.37           | 83.11    | 83.48    | 83.29    |
| Mixed forest                                           | 40.78             | 63.27    | 53.42    | 57.93    | 56.03           | 74.1     | 69.67    | 71.82    |
| Temperate or subpolar shrubland                        | 49.67             | 65.29    | 67.5     | 66.38    | 48.61           | 70.48    | 61.04    | 65.42    |
| Temperate or subpolar grassland                        | 66.33             | 78.9     | 80.63    | 79.76    | 81.44           | 90.12    | 89.43    | 89.77    |
| Sub-polar or polar shrubland-lichen-moss               | 21.68             | 55.47    | 26.24    | 35.63    | 17.62           | 54.06    | 20.73    | 35.04    |
| Subpolar or polar grassland-lichen-moss                | 42.36             | 57.97    | 61.13    | 59.51    | 21.25           | 59.3     | 24.87    | 35.04    |
| Subpolar or polar barren-lichen-moss                   | 37.76             | 53.65    | 56.04    | 54.82    | 37.27           | 66.13    | 46.07    | 54.31    |
| Wetland                                                | 58.08             | 73.75    | 73.21    | 73.48    | 75.76           | 85.78    | 86.64    | 86.21    |
| Cropland                                               | 87.84             | 92.61    | 94.46    | 93.53    | 93.12           | 95.77    | 97.11    | 96.44    |
| Barren land                                            | 72.2              | 82.01    | 85.79    | 83.85    | 70.79           | 84.13    | 81.7     | 82.9     |
| Urban and built-up                                     | 43.61             | 63.08    | 58.56    | 60.74    | 57.79           | 74.52    | 72.01    | 73.24    |
| Water                                                  | 86.36             | 94.45    | 90.98    | 92.68    | 88.25           | 95.04    | 92.51    | 93.76    |
| Snow and ice                                           | 58.75             | 67.31    | 82.21    | 74.02    | 69.57           | 86.6     | 77.96    | 82.05    |
| Cloud                                                  | 97.58             | 97.76    | 99.81    | 98.78    | 91.51           | 94.81    | 96.33    | 95.56    |

Table 7  Global assessment of the models trained on the Landsat 5/7 Prairie and Landsat 8 data from the Lake Winnipeg watershed.

| Evaluation metric          | Landsat 5/7       |          | Landsat 8       |          |
|----------------------------|-------------------|----------|-----------------|----------|
| OA                         | 80.66             |          | 88.04           |          |
| Average                    |                   |          |                 |          |
| CSI                        | 55.19             |          | 62.39           |          |
| Precision                  | 68.72             |          | 79.11           |          |
| Recall                     | 67.35             |          | 71.26           |          |
| F1-score                   | 67.63             |          | 74.07           |          |
| Weighted average           |                   |          |                 |          |
| CSI                        | 71.51             |          | 82.68           |          |
| Precision                  | 82.32             |          | 89.92           |          |
| Recall                     | 82.44             |          | 90.06           |          |
| F1-score                   | 82.33             |          | 89.97           |          |
5 Conclusion

In this study, multiple NNs and extensions were trained on the southern Manitoba using the NALCMS dataset as labeled data. The results have shown that the best model variation is the VGGNet encoder with modified U-Net decoder and OS fixed at 4 that reached 92.4% OA. Six different extensions were applied to the network, where most of them did not provide noticeable improvement both individually, and in combinations. Individual LULC class performance is highly dependent on the number of pixels representing that class; in general, the more pixels, the better the classification accuracy. Additionally, the best model variation was trained on the larger Landsat 5/7 Prairie datasets which resulted in a 80.66% OA; an accuracy that is better than previously reported in the literature but equivalent to the performance of ArcGIS/Pro deep learning package. Also, the same model was trained on the Landsat 8 Lake Winnipeg dataset, where performance reached 88.04% OA, as we speculate, due to higher radiometric resolution. The Landsat 8 data increased the OA by 11.44% compared to Ref. 17 and is 6% to 8% higher than any other deep learning model present in the ArcGIS/Pro package that solves an equivalent problem on a similar satellite, and is on the level of the state-of-the-art DL models designed for datasets with a lower amount of classes. This shows that it is possible to produce fairly accurate NALCMS LULC maps from Landsat 5/7 and 8 sensors for 37 years of archived data and also future Landsat 8 data.
One of the limitations of DL is the amount of labelled data required to generated land cover map products. Thus, one of the most important directions of research in the future is to find a semisupervised solution to produce accurate LULC maps, such as examination of unsupervised domain adaptation architectures.\textsuperscript{59,70} Also, NNs can be trained on high-resolution satellite images to provide sharp and more detailed LULC maps. Furthermore, a multisensor model can be considered by training a network on multiple datasets from different sensors at once, making a more generic and versatile model, which can be used to develop LULC maps from the different sensors without prior training.

References

1. P. Treitz and J. Rogan, “Remote sensing for mapping and monitoring land-cover and land-use change,” \textit{Prog. Plan.} \textbf{61}(4), 269–279 (2004).
2. D. Lu and Q. Weng, “A survey of image classification methods and techniques for improving classification performance,” \textit{Int. J. Remote Sens.} \textbf{28}(5), 823–870 (2007).
3. L. Ma et al., “Deep learning in remote sensing applications: a meta-analysis and review,” \textit{ISPRS J. Photogramm. Remote Sens.} \textbf{152}, 166–177 (2019).
4. A. Mayr et al., “Deeptox: toxicity prediction using deep learning,” \textit{Front. Environ. Sci.} \textbf{3}, 80 (2016).
5. F. Milletari et al., “Hough-cnn: deep learning for segmentation of deep brain regions in mri and ultrasound,” \textit{Comput. Vision Image Understanding} \textbf{164}, 92–102 (2017).
6. P. Pérez de San Roman et al., “Saliency driven object recognition in egocentric videos with deep CNN: toward application in assistance to neuroprostheses,” \textit{Comput. Vision Image Understanding} \textbf{164}, 82–91 (2017).
7. A. Khan et al., “A survey of the recent architectures of deep convolutional neural networks,” \textit{Artif. Intell. Rev.} \textbf{53}(8), 5455–5516 (2020).
8. J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” in \textit{Proc. IEEE Conf. Comput. Vision and Pattern Recognit. (CVPR)}, pp. 3431–3440 (2015).
9. V. Badrinarayanan, A. Kendall, and R. Cipolla, “Segnet: a deep convolutional encoder-decoder architecture for image segmentation,” \textit{IEEE Trans. Pattern Anal. Mach. Intell.} \textbf{39}(12), 2481–2495 (2017).
10. F. P. Luus et al., “Multiview deep learning for land-use classification,” \textit{IEEE Geosci. Remote Sens. Lett.} \textbf{12}(12), 2448–2452 (2015).
11. J. Wang et al., “Transferring pre-trained deep cnns for remote scene classification with general features learned from linear pca network,” \textit{Remote Sens.} \textbf{9}(3), 225 (2017).
12. C. D. Storie and C. J. Henry, “Deep learning neural networks for land use land cover mapping,” in \textit{IGARSS 2018-2018 IEEE Int. Geosci. and Remote Sens. Symp.}, IEEE, pp. 3445–3448 (2018).
13. C. J. Henry et al., “Automated LULC map production using deep neural networks,” \textit{Int. J. Remote Sens.} \textbf{40}(11), 4416–4440 (2019).
14. V. Alhassan et al., “A deep learning framework for land-use/land-cover mapping and analysis using multispectral satellite imagery,” \textit{Neural Comput. Appl.} \textbf{32}, 1–16 (2019).
15. Esri, “Arcgis desktop,” (2020).
16. North American Land Change Monitoring System, \url{http://www.cec.org/north-american-land-change-monitoring-system/}.
17. R. Latifovic et al., “North American land change monitoring system,” in \textit{Remote Sensing of Land Use and Land Cover: Principles and Applications}, C. P. Giri et al., Eds., pp. 303–324, CRC Press (2012).
18. R. R. Colditz et al., “Classifying the land cover of Mexico in the framework of the north American land change monitoring system,” in \textit{Am. Soc. Photogramm. and Remote Sens. Annu. Conf.}, pp. 106–113 (2010).
19. O. Ronneberger, P. Fischer, and T. Brox, “U-net: convolutional networks for biomedical image segmentation,” \textit{Lect. Notes Comput. Sci.} \textbf{9351}, 234–241 (2015).
20. K. He et al., “Identity mappings in deep residual networks,” \textit{Lect. Notes Comput. Sci.} \textbf{9908}, 630–645 (2016).
21. F. Chollet, “Xception: deep learning with depthwise separable convolutions,” in *Proc. IEEE Conf. Comput. Vision and Pattern Recognit. (CVPR)*, pp. 1251–1258 (2017).
22. B. Wang et al., “Feedbackward decoding for semantic segmentation,” arXiv:1908.08584 (2019).
23. T. Karras et al., “Progressive growing of GANs for improved quality, stability, and variation,” arXiv:1710.10196 (2017).
24. L.-C. Chen et al., “Encoder-decoder with atrous separable convolution for semantic image segmentation,” in *Proc. Eur. Conf. Comput. Vision (ECCV)*, pp. 801–818 (2018).
25. M. M. Yapıcı, A. Tekerek, and N. Topaloğlu, “Literature review of deep learning research areas,” *Gazi Mühendislik Bilimleri Dergisi* 5(3), 188–215 (2019).
26. A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” *Adv. Neural Inf. Process. Syst.* 25, 1097–1105 (2012).
27. K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” in *3rd Int. Conf. Learn. Represent.*, Computational and Biological Learning Society, pp. 1–14 (2015).
28. C. Szegedy et al., “Going deeper with convolutions,” in *Proc. IEEE Conf. Comput. Vision and Pattern Recognit. (CVPR)*, pp. 1–9 (2015).
29. K. He et al., “Deep residual learning for image recognition,” in *Proc. IEEE Conf. Comput. Vision and Pattern Recognit. (CVPR)*, pp. 770–778 (2016).
30. Y. LeCun et al., “Gradient-based learning applied to document recognition,” *Proc. IEEE* 86(11), 2278–2324 (1998).
31. E. T. Grozdić, S. T. Jovičić, and M. Subotić, “Whispered speech recognition using deep denoising autoencoder,” *Eng. Appl. Artif. Intell.* 59, 15–22 (2017).
32. M. Al-Ayyoub et al., “Deep learning for Arabic NLP: a survey,” *J. Comput. Sci.* 26, 522–531 (2018).
33. L.-C. Chen et al., “Semantic image segmentation with deep convolutional nets and fully connected CRFs,” arXiv:1412.7062 (2014).
34. F. Yu and V. Koltun, “Multi-scale context aggregation by dilated convolutions,” arXiv:1511.07122 (2015).
35. I. J. Goodfellow et al., “Generative adversarial networks,” *Commun. ACM* 63(11), 139–144 (2020).
36. P. Luc et al., “Semantic segmentation using adversarial networks,” arXiv:1611.08408 (2016).
37. E. Collier et al., “Progressively growing generative adversarial networks for high resolution semantic segmentation of satellite images,” in *IEEE Int. Conf. Data Mining Workshops (ICDMW)*, IEEE, pp. 763–769 (2018).
38. A. Sharma et al., “A patch-based convolutional neural network for remote sensing image classification,” *Neural Networks* 95, 19–28 (2017).
39. D. Pouliot et al., “Assessment of convolution neural networks for wetland mapping with landsat in the central canadian boreal forest region,” *Remote Sens.* 11(7), 772 (2019).
40. D. Pouliot et al., “Assessment of landsat based deep-learning membership analysis for development of from-to change time series in the Prairie region of Canada from 1984 to 2018,” *Remote Sens.* 13(4), 634 (2021).
41. V. Alhassan, “Automated land use and land cover map production: a deep learning framework,” Master’s thesis, University of Winnipeg, Winnipeg, Canada (2018).
42. P. Li et al., “Region-wise deep feature representation for remote sensing images,” *Remote Sens.* 10(6), 871 (2018).
43. J. Sherrah, “Fully convolutional networks for dense semantic labelling of high-resolution aerial imagery,” arXiv:1606.02585 (2016).
44. G. Chen et al., “Symmetrical dense-shortcut deep fully convolutional networks for semantic segmentation of very-high-resolution remote sensing images,” *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* 11(5), 1633–1644 (2018).
45. R. Kemker, C. Salvaggio, and C. Kanan, “Algorithms for semantic segmentation of multi-spectral remote sensing imagery using deep learning,” *ISPRS J. Photogramm. Remote Sens.* 145, 60–77 (2018).
46. B. Chatterjee and C. Poullis, “Semantic segmentation from remote sensor data and the exploitation of latent learning for classification of auxiliary tasks,” *Comput. Vision Image Understanding* 210, 103251 (2021).
47. A. Boguszewski et al., “Landcover.ai: dataset for automatic mapping of buildings, woodlands, water and roads from aerial imagery,” in Proc. IEEE/CVF Conf. Comput. Vision and Pattern Recognit. (CVPR) Workshops, pp. 1102–1110 (2021).
48. F. I. Diakogiannis et al., “Resunet-a: a deep learning framework for semantic segmentation of remotely sensed data,” ISPRS J. Photogramm. Remote Sens. 162, 94–114 (2020).
49. K. Yue et al., “Treeunet: adaptive tree convolutional neural networks for subdecimeter aerial image segmentation,” ISPRS J. Photogramm. Remote Sens. 156, 1–13 (2019).
50. X. Li et al., “Dual attention deep fusion semantic segmentation networks of large-scale satellite remote-sensing images,” Int. J. Remote Sens. 42(9), 3583–3610 (2021).
51. T.-S. Kuo et al., “Deep aggregation net for land cover classification,” in Proc. IEEE Conf. Comput. Vision and Pattern Recognit. (CVPR) Workshops (2018).
52. S. Ren et al., “Faster R-cnn: towards real-time object detection with region proposal networks,” Adv. Neural Inf. Process. Syst. 28 (2015).
53. K. He et al., “Mask R-cnn,” in Proc. IEEE Int. Conf. Comput. Vision (ICCV), pp. 2961–2969 (2017).
54. R. Girshick et al., “Rich feature hierarchies for accurate object detection and semantic segmentation,” in Proc. IEEE Conf. Comput. Vision and Pattern Recognit. (CVPR), pp. 580–587 (2014).
55. N. Otsu, “A threshold selection method from gray-level histograms,” IEEE Trans. Syst., Man, Cybern. 9(1), 62–66 (1979).
56. H. Zhang, L. Zhang, and Y. Jiang, “Overfitting and underfitting analysis for deep learning based end-to-end communication systems,” in 11th Int. Conf. Wireless Commun. and Signal Process. (WCSP), IEEE, pp. 1–6 (2019).
57. D. Ho et al., “Population based augmentation: efficient learning of augmentation policy schedules,” in Proc. of 36th Int. Conf. Mach. Learn. Res., K. Chaudhuri and R. Salakhutdinov, Eds., PMLR, Vol. 97, pp. 2731–2741 (2019).
58. C. Shorten and T. M. Khoshgoftaar, “A survey on image data augmentation for deep learning,” J. Big Data 6(1), 1–48 (2019).
59. M. Abadi et al., “Tensorflow: large-scale machine learning on heterogeneous distributed systems,” arXiv:1603.04467 (2016).
60. D. P. Kingma and J. Ba, “Adam: a method for stochastic optimization,” arXiv:1412.6980 (2014).
61. F. W. Lancaster, Evaluation of the MEDLARS Demand Search Service, US Department of Health, Education, and Welfare, Public Health Service (1968).
62. C. J. Van Rijsbergen, “A new theoretical framework for information retrieval,” ACM Sigir Forum 21(1-2), 23–29 (1986).
63. J. T. Schaefer, “The critical success index as an indicator of warning skill,” Weather Forecast. 5(4), 570–575 (1990).
64. P. J. Roebber et al., “Improving snowfall forecasting by diagnosing snow density,” Weather Forecast. 18(2), 264–287 (2003).
65. H. Lu et al., “Half-U Net: a simplified U-Net architecture for medical image segmentation,” Front. Neuroinf. 16, 911679 (2022).
66. S. Bianco et al., “Benchmark analysis of representative deep neural network architectures,” IEEE Access 6, 64270–64277 (2018).
67. J. M. Johnson and T. M. Khoshgoftaar, “Survey on deep learning with class imbalance,” J. Big Data 6, 1–54 (2019).
68. N. Verde et al., “Assessment of radiometric resolution impact on remote sensing data classification accuracy,” Remote Sens. 10(8), 1267 (2018).
69. E. Tzeng et al., “Adversarial discriminative domain adaptation,” in Proc. IEEE Conf. Comput. Vision and Pattern Recognit. (CVPR), pp. 7167–7176 (2017).
70. B. Benjdira et al., “Unsupervised domain adaptation using generative adversarial networks for semantic segmentation of aerial images,” Remote Sens. 11(11), 1369 (2019).

Biographies of the authors are not available.