ISPRS Journal of Photogrammetry and Remote Sensing 200 (2023) 106–119

Contents lists available at ScienceDirect

ISPRS Journal of Photogrammetry and Remote Sensing

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

Harvesting the Landsat archive for land cover land use classification using deep neural networks: Comparison with traditional classifiers and multi-sensor benefits

Giorgos Mountrakis *, Shahriar S. Heydari

Department of Environmental Resources Engineering, State University of New York, College of Environmental Science and Forestry, 1 Forestry Drive, Syracuse, NY 13210, United States

ARTICLE INFO

Keywords:
Deep neural networks
Recurrent network
Convolutional network
Long Short-Term Memory
Landsat
Random Forest

ABSTRACT

The Landsat archive, with a multi-decadal global coverage is a prime candidate for deep learning classification methods due to the large data volume. Local studies have evaluated deep learning methods on Landsat observations. However, these models often saturate at high accuracies due to limited reference dataset size thus do not fully explore the potential of deep classifiers. Furthermore, no provisions are taken to investigate algorithmic performance of challenging classification areas. To address these shortcomings in this research, Landsat 5, 7 and 8 observations were combined within the continental United States to create one of the largest to date reference dataset containing about 21 million labeled annual temporal sequences. Difficult to classify reference samples were isolated by examining labelsin the immediate vicinity. Long Short-Term Memory (LSTM) and Convolutional Neural Networks (CNN) deep learners were integrated to capture temporal and spatial relationships, respectively. Classification mapping accuracy was contrasted with a commonly implemented large-scale mapping method, the Random Forest (RF).

Results indicate substantial classification improvements of deep learning methods (DLMs) over the RF. These improvements are more pronounced on challenging to classify pixels in heterogenous areas. RF classification accuracy reaches about 70% on average, while DLMs are at 86%-95% range, depending on model architecture. Grass and bare land classes show the highest accuracy improvements, from 65.5% and 63.5%, respectively for the RF to the 79.4%-96.3% range for the DLMs. Our work also examined the practical value of having two, instead of one, Landsat sensors. Results indicate substantial classification increases (7%-10% in average F1 accuracy) suggesting that having two concurrent Landsat sensors is important not only for redundancy but also for improved mapping capabilities.

1. Introduction

Land Cover / Land Use (LCLU) mapping is the process of compiling geographical data and creating thematic maps to delineate different land regions and assign desired labels to them based on features that make up the ground and their intended use. LCLU mapping has direct applications in disaster response, natural resource management, and human-nature interactions (Giri 2016, chap. 1). LCLU maps are also essential for biodiversity studies (Pimm et al. 2014), they guide forest management (Erb et al. 2018) and uncover energy use patterns (Güneralp et al. 2017). In addition to these direct applications, the effect of land cover and land use change (LCLUC) on climate by altering heat fluxes, surface radiation balance, and greenhouse gas fluxes is an active field of study, as promoted by higher concerns on climate change (Pongratz et al. 2021). Clearly, availability of highly accurate, multi-scale historical and current LCLU maps is an essential need for socially important environmental studies. Multiple large scale LCLU mappings efforts have relied on satellite observations. A review by Grekousis et. al (2015) discusses 21 global and 43 regional land cover mapping products covering spatial resolutions from 30 m to 1 km using Landsat, MODIS, MERIS, and other satellite platforms. Pérez-Hoyos et al. (2017) also review seven global land cover maps for cropland classification. More recently, Liu et al. (2021) inspected and compared three popular global land cover products and other thematic global maps.

Deep learning methods have been used for more than a decade in many domains such as computer vision, speech recognition, and natural

* Corresponding author.

https://doi.org/10.1016/j.isprsjprs.2023.05.005
Received 20 September 2022; Received in revised form 29 March 2023; Accepted 4 May 2023
Available online 15 May 2023

language processing (Deng, 2014; Ahmad et al, 2019). Deep learning methods have also found their way in several remote sensing tasks, including image pre-processing, scene classification, pixel-based classification, image segmentation, and target detection (Zhang et al., 2016; Ma et al., 2019; Zhu et al., 2019). Scene classification typically assigns a single label to an image patch. Some demonstrations of scene mapping exist, for example X. Zhang et al. (2021), Rousset et al. (2021), Helber et al. (2019). Another popular implementation of deep learning methods involves transfer learning, the process by which a pre-trained network is fine-tuned with application- and site-specific training data. Transfer learning methods have been mostly applied on scene-based classifications, because the large pre-trained networks originate from computer vision tasks aimed for scene classification or object detection. Example works include Pires de Lima and Marfurt (2019), González-Vélez et al (2022) and B. Zhao et al. (2017).

As the focus of this study is on deep learning methods in pixel classification based on medium-resolution (Landsat) data, we will not elaborate more on scene classification and concentrate on pixel classification throughout the rest of this section. Pixel-based land cover / land use mapping provides a separate label for each image pixel. In addition to the individual pixel's spectral data, other data dimensions or techniques are widely used to enhance the classification. Specific deep learning methods exist targeting the incorporation of spatial information (e.g., using convolutional neural networks), temporal information (e.g., using recurrent neural networks), or spatial and temporal integration.

Adding spatial information is done in a variety of forms, such as flattening spatial information (converting a 2-D spatial neighborhood around a pixel to a 1-D array) and feeding a recurrent network with a sequence of flattened spatial-spectral data (Sharma et al., 2018). Example CNN works include studies in China (X. Zhao et al., 2019), United States (J. Wang et al., 2017), Canada (Alhassan et al., 2020), Iran (Garajeh et al., 2022) along with global efforts (Corbane et al., 2021, Karra et al., 2021). Direct comparisons with non-deep classifiers such as Random Forests (RF) have explicitly quantified accuracy improvements. Y. Wang et al. (2021) used a CNN network enriched with additional processing modules to classify crop types in several US states using Sentinel-2 data and achieved 97.8% accuracy, while RF accuracy was 95%. Saadeldin et al. (2022) used a deep convolutional network to grade grazing land use intensity in Ireland within three classes using Sentinel-1 and Sentinel-2 data, and achieved an accuracy of 92.8%, compared to 84.8% accuracy from RF. Jamali and Mahdianpari (2022) used a complex multi-model deep network to map wetland sites in Canada based on Sentinel-1 and Sentinel-2 data and reported accuracy of 92.3% while an RF classifier reached 91.5%. The above examples are just a few of the very active line of research using various forms of convolutional networks to conduct pixel classification.

Recurrent neural networks (RNNs) are typically used for processing time-series and lengthy temporal data. RNNs have entered remote sensing literature recently (see Lyu et al., 2016) for change analysis, and then found applications in land cover classification with promising results. They are of special interest when the temporal transitions are essential in class identification (e.g., crop classification). For example, RNN-based crop type classification by Rußwurm and Körner (2017) on Sentinel-2 data achieved overall accuracy of 84.4%, and land cover classification by Sun et al. (2019) on Landsat data reached overall accuracy of 89%. Lin et al. (2022) also used a design based on Long Short-Term Memory (LSTM) modules on Sentinel-1 data and reported accuracy of 98.3% in rice paddies mapping in the U.S. In another research, B. Chen et al. (2022) used a bidirectional LSTM design on Sentinel-2 data to classify crop type in a region in China and achieved overall accuracy of about 97%. An example of comparison with a RF classifier is offered by Campos-Taberner et al. (2020), where a bidirectional LSTM implementation on Sentinel-2 data obtained overall accuracy of 98.7% for crop type classification, while the best non-deep classifier was a RF with accuracy of 94.9%.

Of particular interest is the integration of spatial and temporal information within the classification process. Combining CNN and RNN methods can be executed in many ways. One popular approach is to place a convolutional neural network before the RNN step. This idea has been applied to crop type classification in Pelletier et al. (2019) using a stack of Formosat-2 images. The CNN implementation can be integrated within the LSTM cells, as Rußwurm and Körner (2018) used to process Sentinel-2 data for 17-class crop type classification with overall accuracy of 90%. The CNN part can also be placed after the recurrent part as demonstrated by Mazzia et al. (2019) who reported overall accuracy of 96.5% using Sentinel-2 data for crop type classification. Finally, the CNN part can be placed in parallel to the recurrent network as used by Interdonato et al. (2018), where they passed input image stacks through parallel RNN and CNN branches and aggregated the branches output in one data vector and classified the result. Using Sentinel-2 data, they reported overall accuracies of 86.1% and 96.8% for two land cover classification case studies. Parallel CNN and RNN architectures have also been applied in multi-sensor fusion tasks. For example, Landsat, high-resolution imagery via the National Agriculture Imagery Program (NAIP) dataset, climate data via the PRISM dataset, and terrain topography data were fused by Chang et al. (2019). Thorp and Drajat (2021) also used different LSTM and CNN combinations to map paddy rice fields in Indonesia using Sentinel-1 and Sentinel-2 data but achieved similar test accuracy of about 76% in their different settings. Masolele et al. (2021) used various models of spatial, temporal, or spatio-temporal deep models (CNN and LSTM-based designs) over Landsat 5/7 data to assess land use after deforestation for selected areas in different continents and confirmed higher accuracy of hybrid spatio-temporal models.

Another problem that specially hinders development of deep models is the availability of large reference datasets. Three notable large datasets are available for satellite-based land classification, namely BigEarthNet, LUCAS, and LCMAP. BigEarthNet contains approximately 590 K land patches of 1.2kmx1.2 km, each patch labelled with multiple land covers (Sumbul et al., 2021). Due to the variably patch size it is not usable for pixel-based classification. Two large, pixel-based, reference datasets are available for medium resolution imagery. The Land Use/Cover Area frame Survey (LUCAS) dataset contains a point survey for more than 1.1 million point (Andrimond et al., 2020) that combines photointerpretation with ground surveys. The Land Change Monitoring, Assessment, and Projection (LCMAP) reference dataset was recently released by the U.S. Geological Survey. The dataset contains 25,000 points with land cover/use labels assigned annually during the period of 1984–2018 for a 30 m x30m area (Pengra et al., 2020).

The potential of deep learning methodologies to advance LCLU mapping has been extensively demonstrated. Our study seeks to quantify further these accuracy improvements through an assessment over a newly developed reference dataset across the continental U.S. Our work offers two key distinctions: i) our reference dataset is substantially larger that previous efforts, thus allowing data-hungry deep learning methods to reach their full potential, and ii) we offer an explicit assessment on difficult to classify pixels – defined as pixels where LCLU class spatial transitions occur – thus amplifying algorithmic performance differences in the most challenging to classify pixels.

2. Data

2.1. Study area

Our study area was the entire conterminous United States. We received 2717 land cover samples for 10 km × 10 km blocks for each ecoregion from USGS, which was originally produced for the USGS Land Cover Trends Project (https://www.usgs.gov/centers/wgsc/science/land-cover-trends). The blocks composed of 333 × 333 pixels (at 30 m nominal ground resolution) using the Albers Conical Equal Area projection and were validated for the year 2000 (note we manually

verified them for 2005–2019 – see section 3.1). Each pixel was labeled according to a modified Anderson classification system to designate the pixel's dominant land cover/use. Eleven classes were assigned: Water, Developed/Urban, Mechanically Disturbed (human-induced disturbances), Barren, Mining, Forests/Woodlands, Grassland/Shrubland, Agriculture, Wetland, Nonmechanically Disturbed (disturbances caused by natural causes such as caused by wind, floods, fire, animals), and Ice/Snow.

One representative block for each of the 84 level III EPA ecoregions was selected for further investigation and refinement (Fig. 1). These ecoregions guided sample stratification as defined in https://www.epa.gov/eco-research/ecoregions, ecoregions are areas where ecosystems and the type, quality, and quantity of environmental resources are generally similar. The selection criteria for a representative block from each ecoregion included high class diversity and balanced distribution of land cover types.

2.2. Data types

Landsat Surface Reflectance and topographic data were the data sources. All datasets were freely available on the Google Earth Engine platform, which was used for data access and dataset generation. Landsat surface reflectance Tier 1 data was used in this study, which has already been corrected for atmospheric errors. Landsat *radsat_qa* and *pixel_qa* quality bits were also used for each pixel to identify radiometric saturation and cloud or cloud shadow conditions (medium or high confidence) and remove those pixels. The Landsat 7 errors due to SLC failure have already been processed by the Google Earth Engine and those pixels were masked. Landsat 5, 7, and 8 sensors were integrated and six Landsat bands were included: Blue, Green, Red, NIR, SWIR1, and SWIR2 bands. The Shuttle Radar Topography Mission (SRTM) digital elevation data V3 product as provided in Google Earth Engine catalog was used to extract elevation, slope, and aspect fields. These three variables were considered static over the entire study period.

3. Methods

The methods section presents details for dataset generation (both reference and model input data), simulation framework, and model architecture. We also discuss benchmark algorithms, and performance evaluation criteria.

3.1. Dataset generation

A subset of all available pixels within the 84 EPA blocks was extracted. Within each block, possible changes within each pixel's land cover were visually inspected using Google Earth high-resolution imagery and pixels with stable land cover over a long time period were selected. This period was generally considered to be 2005–2019 (including both start and end year) but may vary based on availability of high-resolution imagery for each location. Some pixels were dropped due to uncertainty and/or instability of land cover type, so our reference maps were patchy and not contiguous. To increase confidence on the produced dataset all pixels were visually inspected.

3.1.1. Assigning class labels

Possible land cover types were reduced from 11 in the original USGS dataset to 7. This included water, developed, grass/shrub, forest, bare, agriculture, and wetland classes. No ice/snow class was present in our selected points. Class definitions are provided in Appendix A, along with further discussion on low quality or missing data, mixed pixels and transitions (for example between forest and grassland), dynamic boundaries (such as in wetlands), and class priorities. Briefly, the labeling process followed a progressive rule-based approach. Highest priority was given to developed areas. If at least 20% of a pixel's area was considered developed, then that pixel was assigned to the developed class independently of the rest of its content. The next priority was given to the agricultural fields with a similar 20% minimum pixel area coverage. If a pixel was not assigned with the above two rules, then a land cover type was selected using a simple majority rule. Further rules were developed to distinguish farm from grassland, grass or forest from

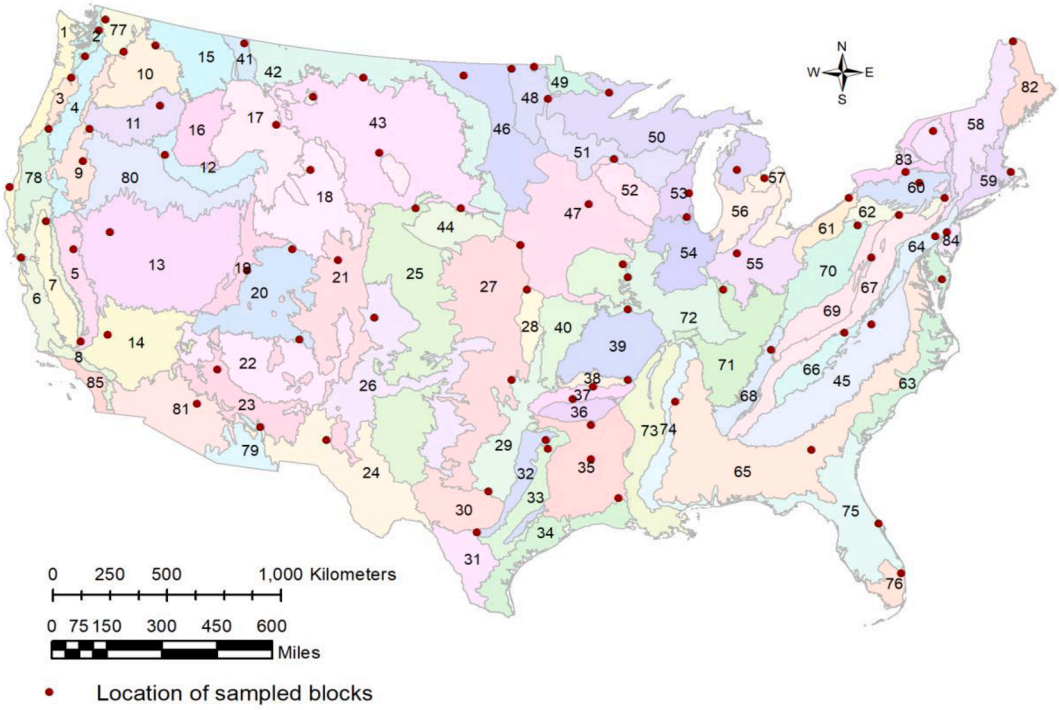

Fig. 1. Level III ecoregions in the conterminous United States (source: https://www.epa.gov/eco-research/level-iii-and-iv-ecoregions-continental-united-states) and selected blocks (red circles). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

wetland, wetland from water, bare from grassland, etc. A summary of required steps for data processing and overall workflow is given in Appendix B.

3.1.2. Pixel selection and annual sequence generation

Upon completion of class labeling, qualified pixels for each of the 84 blocks varied between 35,000 up to 100,000 valid pixels. Also, time spans ranged from 8 to 15 years. To keep simulation times practical, about 7,000 to 50,000 pixels in each block were initially selected to reach a total of about 1.6 M (million) pixels. Details on block class distribution are provided in Appendix C. Class distribution varied considerably with wetland and bare land classes underrepresented. Early tests showed that wetland and developed classes were more challenging to classify accurately, and wetland and bare classes were low in frequency. Therefore, to create the final dataset all bare and wetland pixels were included. Also, a larger proportion of developed class pixels (compared to the other classes) was selected. After generating annual sequences for each available year for each pixel, the final dataset contained approximately 21 M annual sequences with the class distribution presented in Table 1.

Each sample of the 21 M annual sequences corresponded to a specific pixel location and represented available data for a particular year. The length of each Landsat sequence varied from pixel to pixel (e.g., due to clouds) and block to block; the highest number was 98 observations per year (mean of 50) and corresponded to pixels covered by multiple adjacent Landsat scenes. These different sequence lengths required a zero-padding process, where extra feature records with zero values were added to the sequence to make all annual sequences having an equal number. The features to include in each sequence varies based on the selected model type, as described in section 3.3.

3.1.3. Calibration and validation data generation

The dataset was divided into calibration (for model optimization) and validation (for accuracy assessment) partitions. The calibration dataset was further divided into training and testing. Training sets were used to train the neural network during specified training epochs, while its companion testing data was used after each epoch during training to evaluate the model performance on unseen data and stop the training when it is no longer useful for generalization (i.e., prevent overfitting of the model).

The validation dataset, where all accuracy reporting was conducted on, consisted of 4/35 (about 11.5% or 2.4 M annual sequences) of the reference data, with the calibration data using the remaining data. This ratio between validation and calibration is typical for complex model development, it is usually in the 10%-15% range assuming large sample dataset (as it was in our case). The validation data were spatially disjoint from the calibration data, in essence a given pixel location would provide data only for validation or calibration but not both. For the calibration data 28/35 (80% or 16.9 M sequences) was used for training and the rest (8.5% or 1.8 M sequences) for internal testing during model development. The calibration dataset was sampled N times to create N calibration sets. The reason was twofold: to reduce neural network performance variance that is caused by inherent randomness in its training, and to enhance our estimation of model performance on unseen data. The best value of N depends on the level of confidence and the acceptable generalization error, but heuristically $N = 10$ is an acceptable norm (Iyer and Rhinehart, 1999). We determined it more practically by running the network N times, looking at the average performance, ranking the performances for different configurations, and observing when this ranking stabilized. We found that $N = 8$ was a good start for training sets of about 500,000 samples but when the size of data or network parameters increased, N can be lowered because the model variance also decreased.

3.2. Model training and optimization

Deep networks offer advanced modeling capabilities but also have high complexity. This poses a significant challenge as almost infinite architecture combinations (e.g., nodes per layer) and node parameters (e.g., activation functions) exist. Despite our considerable computational resources and the large reference dataset, practical limitations dictated a step-by-step approach as simultaneous exhaustive parameter search was not possible.

The simulation parameters can be divided in three main groups: 1) input features, 2) network structure (number of layers, neurons per layer, etc.), and 3) network optimization settings (training batch size, optimizer type, learning rate, etc). To keep things manageable, the simulation framework was designed in below four steps:

a) Deciding on the best setting of network optimization parameters using a fixed network configuration, input features, and input data size.
b) Finding the best combination of possible input features using the same network in step (a).
c) Increasing network complexity step by step until there is no significant improvement in performance using the best input features found previously.
d) Final model adjustments by revisiting optimization parameters.

We considered the number of network parameters as a measure of network complexity, and assume the size of input data set should be multiple times bigger than the number of network parameters. For small network size, a factor of 10 times might be reasonable and feasible but as the network grows bigger, keeping this ratio become impractical because it increases the training time proportionally. Therefore, the regularization will be required at steps (c) and (d) method (we chose dropout method) in all of our network implementations to prevent network from being overfit without the need for too much input data. Note that each single run in the above steps (a-d) is actually a set of N simulations exactly under the same configuration but with different input calibration set, as described in the previous section, and the results from all of these N simulations are represented by a single number to compare this run to the others.

3.3. Model candidates

The general schematic of our approach is shown in Fig. 2. To assess the value of temporal and/or spatial information three network architectures were tested. The first deep learner only examines individual pixel temporal information using a Long Short-Term Memory network (T-LSTM). The second network (ST-LSTM) adds expert-selected spatial neighboring features to the T-LSTM. The third network (C-LSTM) adds to the ST-LSTM automatically generated features using a convolutional neural network. We recognize that these may not be the optimal model candidates, as more advanced methods are continuously developed, for example multi-head self attention CNNs. Our intent is to evaluate here typical starting deep learning methods used as baseline for such implementations, with more advanced methods reserved for future work as the deep learning field matures further within the remote sensing community.

Table 1
Final dataset class distribution.

Water	Developed	Grass/Shrub	Forest	Bare	Agriculture	Wetland	Total
1,407,689	7,131,290	3,657,994	2,574,143	1,091,321	3,888,984	1,430,939	21,182,360

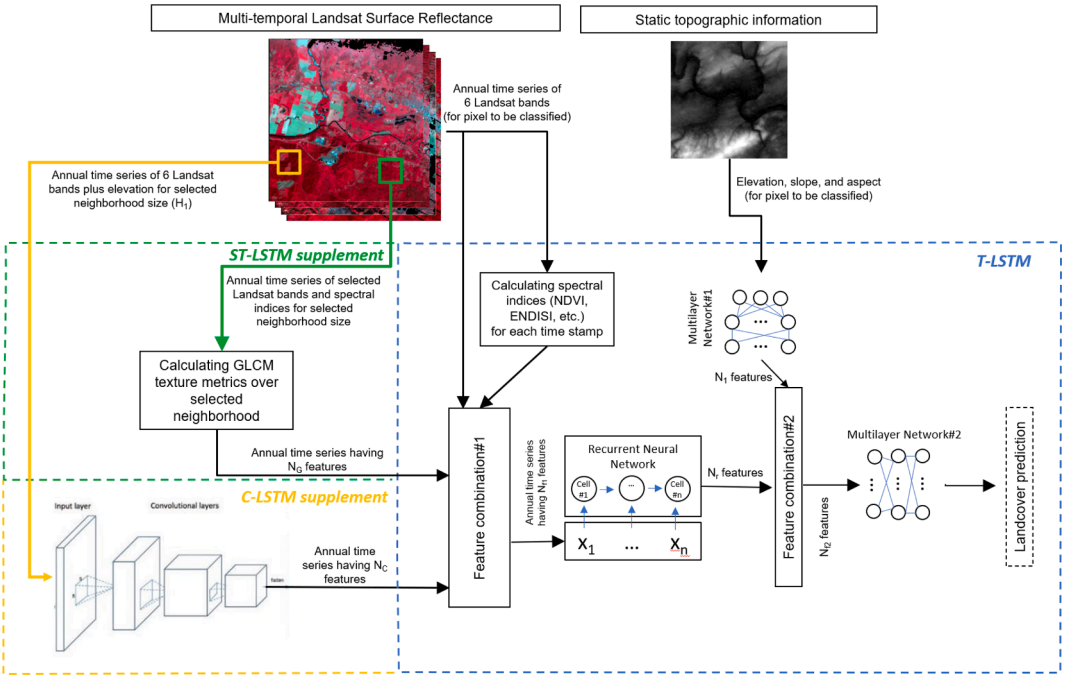

Fig. 2. Designed system architecture. T-LSTM shows the building blocks of the basic model. The ST-LSTM model adds the ST-LSTM supplement to T-LSTM, and the C-LSTM model adds both ST-LSTM and C-LSMT supplements to the model. Resulting architectures are presented in Table 2.

The base data in all models were sequences of Landsat data and static topographic data. Each annual sequence contained several temporal records. Each record represented a specific observation day and sensor. For the T-LSTM the annual Landsat sequence was fed to a recurrent network (Fig. 2). Static topographic data were in the form of 1-D vector and for each pixel, there was one vector of static features corresponding to one annual sequence of Landsat-based features. The static features were processed by a standard multilayer neural network. Output features from the recurrent and the standard neural networks were concatenated and fed to a second multilayer network for further processing with the latter producing the assigned land cover to the input pixel.

In the case of the ST-LSTM the only difference was that the recurrent network was also fed with expert-selected patch statistics from neighboring pixels. For the C-LSTM, in addition to the expert-selected spatial features, features selected automatically from a convolutional neural network were also included as inputs for the recurrent network (Fig. 2).

3.3.1. *Temporal LSTM (T-LSTM)*

This was the starting base model that processed multi-temporal spectral data. There were eleven base variables for each temporal observation for each pixel: Day-of-year (DOY), sensor type (Landsat 5/7/8), six Landsat surface reflectance values, and three topography variables (elevation, slope, aspect). In addition, eight different spectral indices were considered as candidates, which have been reported in the literature to be useful for identification of different ground features such as vegetation, water, built-up area, bare soil, and soil wetness. The list of the reviewed indices and their equations is provided in appendix E. Among all of these indices, our experiments showed ENDISI (Chen et al., 2019) significantly improved network performance. All spectral index calculations were done in Google Earth Engine while extracting Landsat data. From the network architecture perspective, different cell types proposed as building blocks for recurrent neural networks were considered. LSTM (Long Short-Term Memory) and GRU (Gated Recurrent Unit) are the most popular types. LSTM was selected as initial experiments showed it was performing slightly better, and it has more trainable parameters.

3.3.2. *Spatio-temporal LSTM (ST-LSTM)*

As shown in Heydari and Mountrakis (2018), relying solely on spectral data is not sufficient to take advantage of deep neural network classifiers as they perform similarly to other classifiers. The next model added spatial data dimension by including texture features. Two spatial information extraction methods were considered: Gray-Level Co-occurrence Matrix (GLCM) and Local Binary Patterns (LBP). The LBP method finds local spatial patterns around a pixel and codifies it to a corner, edge, or middle of a homogeneous area (Ojala et al. 2002). As our initial tests did not show any benefit for LBP over GLCM, it was not considered further.

The GLCM method generates a co-occurrence matrix from any image band of interest from which multiple metrics are calculated that are used to describe the texture around the pixel (Hall-Beyer, 2017a). GLCM produces various texture metrics for each pixel and can be organized in three main groups: contrast group, orderliness group, and descriptive statistics. Each group contains several metrics, which was tried individually and in combination. There is no clear best metric, since this depends on the application and GLCM parameters. Hall-Beyer (2017b) looked at this issue for a classification application based on Landsat data and recommended choosing Mean/Correlation (for general texture identification), Contrast/Dissimilarity (helpful for classes containing edge-like features), and Entropy (for more detailed texture study). Our analysis resulted in selection of four metrics: dissimilarity, entropy, mean, and variance. GLCM generation requires specifying two other parameters: GLCM window size and quantization level. The quantization level was fixed at 64. The above four GLCM features were generated for two window sizes (radius) of 5 and 15 pixels to represent different spatial scales. GLCM generation also requires picking a base band to do the spatial calculation on. Past studies have used either one of visual bands, the NIR band, or a generated band such as a principal component. In our case, selection of final GLCM features and base bands was the result of experimentation. The performance of a sample network was tested when fed by GLCM features generated from all potential bands and indices. The final GLCM base bands were two Landsat bands (blue, NIR) and two spectral indices (DD, ENDISI). At the end, we had 32 GLCM features representing the combinations of the four base bands, the four metrics and the two neighborhood scales.

For simplicity, a three-part name was adopted to designate each GLCM feature. For example, ENDISI_ent_15x64 denotes the GLCM entropy metric generated based on ENDISI band using a window size of 15 and quantization level of 64. All GLCM calculations were done in Google Earth Engine with available functions.

3.3.3. CNN and Spatio-Temporal LSTM (C-LSTM)

The last and most complete model was built upon the ST-LSTM model by supplementing the expert-selected spatial features (GLCMs) with computer-generated convolutional features. In this model the CNN block in Fig. 2 was utilized and the spatial features generated by it were added to the input features of the recurrent network. The combination of six Landsat surface reflectance bands plus the elevation layer were selected as inputs to the CNN. For each sample, the neighborhood data for the chosen bands was extracted. Then standard 2-D convolutional filters without padding were used for conducting convolution to generate a 1-D vector as output. For example, for an input neighborhood of size 5, a 3x3 filter in the first convolutional layer reduced the 5x5 input to 3x3 (5–3 + 1 = 3), and then a 3x3 filter in the second convolutional layer reduced it to a spatial size of 1x1. The rest of network was the same as the Spatio-Temporal LSTM.

3.4. Benchmark algorithms

The Random Forest (RF) method, a popular classifier for large scale classification products (e.g. the NLCD) was used as baseline for comparison to the deep learning models. One important distinction between the RF models (baseline models hereafter) and the proposed deep models is that in our designed network, the data has a built-in temporal dimension and is presented to the deep networks in the form of temporal sequences. This is not the case for the baseline models, which are not recurrent in nature. As each yearly record in our dataset contains on average about 50 timestamps, combining all time stamps of one year together and providing them as simultaneous input for baseline models was not feasible due to the large data volume, RF methods are not designed for such high dimensionality inputs. Furthermore, it would pose issues for generalizing to other sites as the input dimensionality would need to be kept to a fixed number (RF design limitation), while in reality the dimensionality varies due to variable clear day observations per year. Therefore, one candidate observation was selected from each season of the year as close as possible to the season midpoints (i.e., day-of-year values of 15, 106, 197, and 288). These four timestamps were concatenated to form the input feature vector. For each of the above 4 timestamps, the same variables as the spatio-temporal recurrent neural network model (introduced in the next sections) were collected, including day-of-year, sensor type, six Landsat bands, selected spectral indices, and selected GLCM texture metrics. This composite feature vector was supplemented by topographic variables and the result is used for RF model training and validation. Different parameterization settings were tested to find the best RF model: number of estimators (from 50 to 200), maximum tree depth (30/40/50/no limitation), and minimum leaf size (from 1 to 5).

3.5. Accuracy assessment

3.5.1. Metrics and dependencies

Accuracy reporting was conducted exclusively on the validation dataset. One important consideration in accuracy assessment is the independence of calibration and validation datasets. Spatial independence was ensured through selection of different pixels. However, temporal independence was not enforced as that would significantly limit the dataset sizes. All years were included in both calibration and validation to offer the ability for the models to adapt to any abnormal annual conditions (e.g., extreme weather) and be able to validate it. This is the normal practice in other literature that deal with time dimension, for example change detection using two fixed time stamps. As we seek balanced performance in all classes and overall accuracy is more representative of the performance of the dominant class, we opted to use the F1 metric[1] for each class and then calculated the average of this value over all classes to obtain an aggregate performance measure. Minimum F1 value and overall accuracy were also calculated and reported. Model selection was based on higher average F1. If two models had very close average F1 values, the minimum F1 and overall accuracy were also considered to make a decision. The reported assessment was conducted on the entire validation dataset (2.4 M sequences) unless otherwise specified. It should be noted that each block may have a different share as listed in appendix C. Finally, when permitted by the dataset size, each classifier was simulated multiple times to obtain a higher confidence on its performance.

3.5.2. Spatial edge samples representing challenging classifications

In order to further assess algorithmic performance in challenging to classify pixels, a subset of the validation dataset was created. This validation dataset, called spatial edge samples, aimed at identifying pixels that are at the edges of differing LCLU areas. Each pixel under consideration (center pixel) was contrasted with its adjacent 8 neighboring pixels. We proceeded to count the number of neighbors with different labels than the label of the center pixel. Center pixels were assigned in three categories, having one differing neighbor, two differing neighbors or three or higher differing neighbors. For this process the labels were extracted using the most accurate model (C-LSTM) and the spatial edge condition should be consistent for at least 3 years to limit the effect of misclassifications.

These samples offered a better insight on classification performance as they exclude samples surrounded with the same LCLU class. This limits the inclusion of homogenous areas that often bias accuracy assessment as they tend to artificially inflate accuracy (e.g. center of water bodies, dense forested areas, agricultural parcels). For consistency in our results section, we report accuracy statistics on both the spatial edge samples and the entire dataset.

3.6. Algorithmic development

Development and implementation of our model and data processing steps were conducted on different platforms with all coding using Python. The Tensorflow environment was used for model development and simulation. Data extraction was done via the Google Earth Engine platform, then the data were downloaded locally. It was followed by pixel sampling and calibration/validation datasets generation. Although initial tests and model evaluation were conducted on our local resources, most model training took on a cluster of powerful GPU-enabled nodes (NVIDIA V100) available through NASA's High-End Computing facilities at NASA Ames Research Center. We used up to 56 single-GPU and 28 4-GPU nodes during different stages of model training, comparison, and selection, which also required corresponding data transfers and job scripting tasks. A more detailed description of the algorithmic implementation is provided in appendix B.

4. Results

4.1. Model training and optimization parameters

Due to the large reference dataset size, a dual step training process was followed. Initially, a randomly extracted subset was used to provide general feedback on training parameters – 560,000 samples for training, 140,000 samples for validation, and 280,000 samples for testing.

[1] F1 metric is defined for each class as the harmonic mean of the class precision and recall. It is calculated asF1 = 2 × precision × recall / (precision + recall).

Table 2
Characteristics of selected deep network models.

Model type	T-LSTM	ST-LSTM	C -LSTM
Number of model parameters	52,663	2,297,127	2,685,287
Training parameters			
Batch size	1024	1024	1024
Optimizer	Adadelta	Adam	Adam
Optimizer parameters	Learning rate = 1.0	learning rate = 0.001, AMSgrad = True	learning rate = 0.001, AMSgrad = True
Final network architecture			
Layer structure:CNN layers: (# of filters and neighborhood size) per layer	N/A	N/A	(128,3), (96,3)
LSTM layers: # of cells per layer			
Multilayer network#1: # of neurons per layer	48, 48, 48	320, 320. 320	340, 340, 340
Multilayer network#2: # of neurons per layer	16, 16	64,32	32, 32
	32, 32	256, 256, 256	128, 128, 128, 128
*Dropout regularization**	0.2, 0.2, 0	0.25, 0.1, 0	0.3,0.25,0.05,0.05
Input features**			
T-LSTM	DOY, sensor, Landsat SR 6bands, Topography, ENDISI		
ST-LSTM	DOY, sensor, Landsat SR 6bands, Topography, ENDISI, DD_ent_5x64, ENDISI_ent_15x64, blue_savg_15x64		
C -LSTM	*CNN subnet input*: Landsat SR 6 bands + Elevation *Rest of network*: DOY, sensor, Landsat SR 6bands, Topography, ENDISI, DD_ent_5x64, ENDISI_ent_15x64, blue_savg_15x64		

* Dropout ratios are given as a tuple of numbers and each number belongs to one of the blocks mentioned in the Layer Structure row. If not zero, the dropout ratio will be applied to all layers of that block.
** Abbreviations: DOY (Day Of Year), ENDISI (Enhanced Normalized Difference Impervious Surfaces Index), DD (Drought Distance index).

Five learning rate optimization algorithms were considered: Adadelta, Adagrad, Adam, RMSprop, and SGD as implemented in Tensorflow/Keras. They were tested on a relatively simple network of LSTM (3x48) followed by two multi-layer networks with 16 and 32 nodes respectively. The Adam optimizer (with AMSgrad option set to True) was the best method under various initial learning rate settings. The SGD option resulted in highly variable performance but did not outperform the Adam optimizer. The other options provided performances between the above two, with Adadelta showing almost as good performance as Adam while being quicker.. We choose Adadelta for our main simulations to save time, and when the model architecture was finalized Adam was implemented in the last tuning stages to get the best possible performance.

The following range of parameters was tested to identify the optimal solution:

- Increasing LSTM subnetwork layers up to 6 and number of cells per layer up to 480,
- Increasing D1 subnetwork layers up to 4 and number of neurons per layer up to 128,
- Increasing D2 subnetwork layers up to 4 and number of neurons per layer up to 512,
- Varying dropout ratio after LSTM layers within 0.2–0.4, and after dense layers within 0 – 0.2,
- Adding L1/L2 regularization (as a replacement to dropout or in addition to it) to recurrent and/or dense networks with the regularization parameter ranging from 0.01 to 0.05,
- Switching to Adam optimizer in last steps, and changing optimizer learning rate. For Adadelta, we changed it within 0.5 – 10, and for Adam within 0.001 – 0.035,
- Examined sigmoid, tanh, and relu activations for dense subnetworks (activation function for LSTM subnet cells was fixed to tanh due to specific CUDA-based GPU implementation in our code),
- Tested training batch size within the 256 – 4096 range.

Our testing indicated that performance was not increasing after certain level of network complexity with ~2 M parameters offering a good balance. Dropout showed to be quite useful particularly for improving complex networks generalization, but L1/L2 regularization was not impactful. The Adam optimizer was better than Adadelta, but it became unstable with increasing learning rates.The tanh activation performed better than sigmoid and relu functions, while the batch size of 1024 was the preferred choice.

The final parameters of the selected models are listed in Table 2. Note that the reported network details (e.g., number of layers and cells in each layer) are not unique and other network configurations can provide similar performance. In fact, the variability in reported accuracy was very small (and indeed random, due to the intrinsic nature of neural networks) in some networks with millions of parameters. Therefore, the network details below should be considered as demonstrative. For the RF model different feature combinations were tested. When all features were included, the highest accuracy was obtained. For hyperparameter selection after testing different values for number of trees, tree depths, and minimum leaf sizes we found that classifier performance saturated above 100 trees, and minimum leaf size of 1 and depth of 50 provided best performance for both overall accuracy and average F1 accuracy.

4.2. Comparison of baseline models and deep learning methods

4.2.1. Quantitative comparison

The validation dataset was organized in four different categories to capture model performance across varying levels of classification difficulty. Firstly, the entire validation dataset was used. While this dataset expresses algorithmic performance across all validation pixels results are biased by the large number of pixels that are easy to classify independently of the methodology used. Examples include pixels inside large homogenous areas, such as agricultural parcels, lake bodies, forests and urban centers. To focus on challenging pixels three spatial edge sample groupings were created following the process described in section 3.6. The entire validation dataset included 2,297,679 sample sequences, while the spatial edge dataset identified 735,833 samples having one differing neighbor, 582,241 samples with two differing neighbors, and 468,667 samples with three or higher differing neighbors.

Results are presented in Table 3 along with a targeted comparison between RF and the best performing deep network (C-LSTM) in Fig. 3.

The entire dataset results are reported for completeness; however,

Table 3
Accuracy over the entire dataset and spatial edge samples.

	Entire dataset				Differing neighbors = 1				Differing neighbors = 2				Differing neighbors ≥ 3			
F1 per Class	RF	T-LSTM	ST-LSTM	C-LSTM	RF	T-LSTM	ST-LSTM	C-LSTM	RF	T-LSTM	ST-LSTM	C-LSTM	RF	T-LSTM	ST-LSTM	C-LSTM
Water	98.1%	97.9%	98.9%	99.3%	85.6%	89.8%	93.9%	96.0%	84.2%	88.9%	93.3%	95.6%	83.1%	88.3%	92.8%	95.3%
Impervious	94.5%	92.5%	97.1%	98.1%	70.2%	89.1%	93.9%	96.0%	68.2%	88.3%	93.4%	95.7%	66.8%	87.8%	93.1%	95.4%
Grass	94.2%	91.6%	96.5%	97.4%	65.5%	79.4%	90.2%	92.7%	61.9%	76.5%	88.6%	91.4%	59.7%	74.8%	87.4%	90.6%
Forest	92.8%	94.5%	95.6%	96.9%	75.2%	86.3%	90.1%	92.9%	73.7%	85.2%	89.2%	92.3%	72.9%	84.5%	88.5%	91.8%
Bare	98.4%	95.9%	99.0%	99.4%	63.5%	84.3%	94.0%	96.3%	60.7%	82.6%	93.3%	95.9%	58.2%	81.4%	92.4%	95.4%
Agriculture	93.7%	92.7%	97.2%	98.2%	70.0%	85.0%	93.3%	95.5%	68.8%	84.1%	92.7%	95.1%	68.3%	83.7%	92.3%	94.8%
Wetland	94.7%	92.3%	97.6%	98.3%	69.6%	84.2%	93.7%	95.6%	67.7%	82.9%	93.0%	95.1%	66.7%	82.2%	92.6%	94.8%
Overall accuracy	94.5%	93.1%	97.1%	98.0%	70.3%	86.0%	92.7%	95.0%	68.3%	84.8%	92.0%	94.5%	67.0%	84.1%	91.5%	94.1%

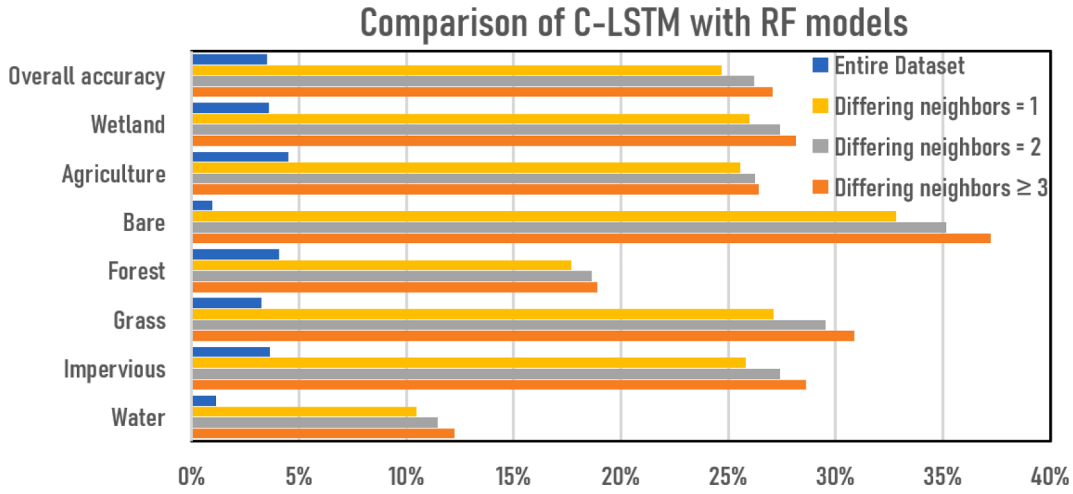

Fig. 3. Random forest classification accuracy loss compared to C-LSTM classifier.

conclusions are difficult to extract as the included pixels are of varying difficulty. In general, though it seems that the combination of spatial and temporal information presented to the RF outperforms the temporal only T-LSTM deep neural network. As a reminder the RF uses 4 temporal scenes per year and various spatial features, while the T-LSTM would typically have longer temporal sequences, however without any neighboring spatial information. On the other hand, it is also clear that when spatial information complements the temporal information presented to a deep learning method, either as expert-defined features as in the ST-LSTM case or automatically extracted through convolution as in the C-LSTM case, deep learning methods considerably outperform the RF benchmark classifier. These improvements are evident across all LCLU classes and while not large in absolute values they are substantial considering the portion of the RF errors they are able to correct.

A clearer algorithmic comparison can be conducted using the spatial edge groups, looking more specifically where different LCLU classes may be directly adjacent. Results here overwhelmingly favor deep learning methods. As expected, all models have reduced accuracy as the heterogenous LCLU neighborhoods become progressively more apparent (differing neighbors from 1 to 3 or more). However, the RF performance takes a major hit decreasing to about 70% on average, even with a single differing neighbor, while deep learners hold at 86% or higher. Grass and bare land classes show the highest accuracy drops at 65.5% and 63.5%, respectively for the RF, but they are in the 79.4%-96.3% range for the deep learners. We should note again here that the reduced RF accuracy is the combined effect of limited multi-temporal support and internal algorithmic limitations.

The spatial edge sample analysis identifies the C-LSTM as the clear winner from the three tested deep learning methods. The bar graph of Fig. 3 contrasts further the best deep learning method (C-LSTM) with the RF benchmark classifier. Even looking at the less restrictive spatial edge samples with only one differing neighbor C-LSTM improvements range from 10.5% for the water class and 17.7% for the forest class to 24.7% or higher for the other classes. The highest improvement can be found with 32.8% for the bare class. These improvements are highly convincing considering the spatial edge dataset size (468,667 samples or higher) and geographic distribution across the continental U.S.. Improvements are also more pronounced with higher differing neighbors suggesting that our hypothesis that deep learning methods are particularly suited for difficult classifications to be true. Further details on error (confusion matrices) and class precision/recall metrics for each case in Fig. 3 are presented in appendix E.

Another important factor in assessing the C-LSTM performance is the consistency on the obtained results. Appendix D presents the minimum and average F1 accuracy of the forty best performing C-LSTM architectures. Results indicate very small variability in classification accuracy, thus enforcing more the validity of the C-LSTM as a reliable classifier.

4.2.2. *Qualitative comparison*

To further compare the classification outputs of the four methods, two sites were selected north of Atlanta, GA (see Figs. 4 and 5 map). Both sites were contained on a single Landsat scene (path 16, row 31). Classification products were created using available Landsat observations for the year 2016. Both sites support the progressive improvements in the Developed class presented in Table 4. While large, developed areas were captured by all methods, isolated pixels, such as roads and houses surrounded by vegetation were more consistently captured by the deep

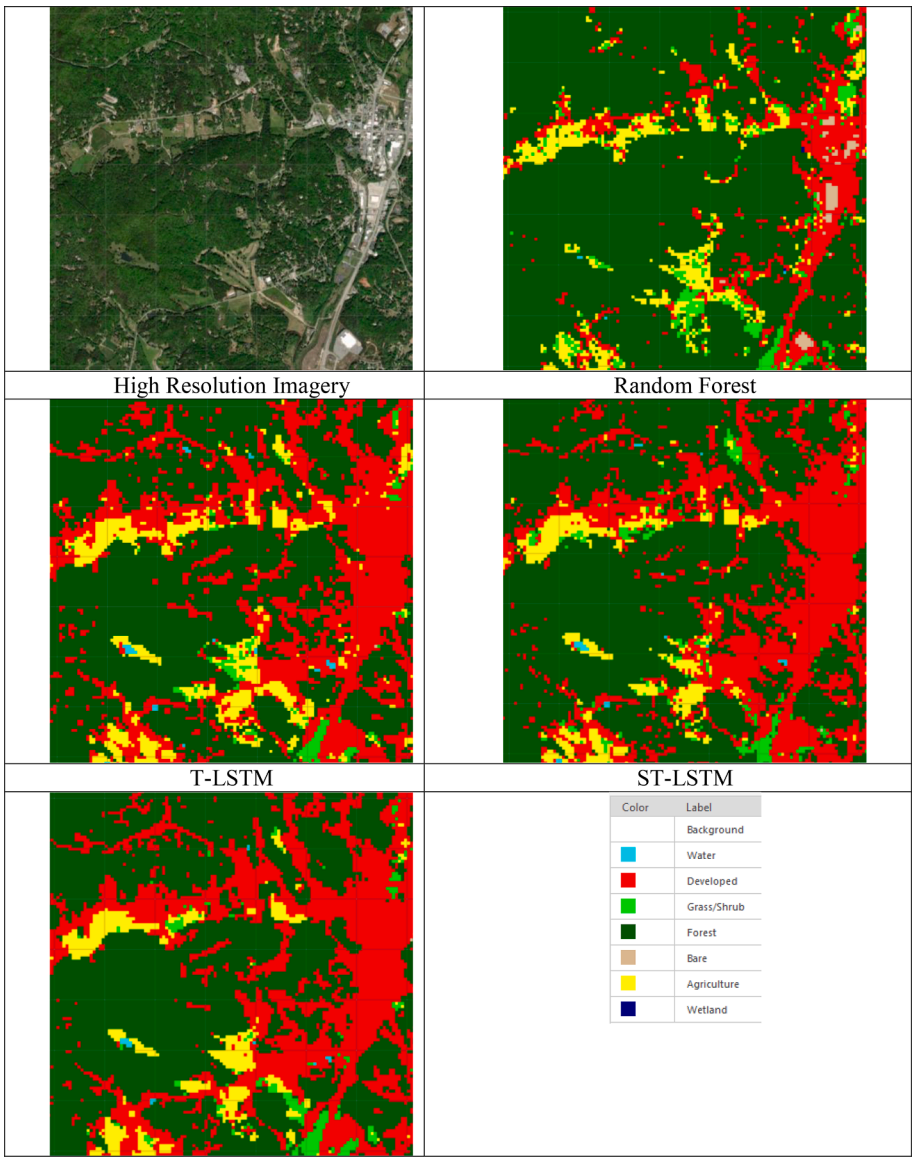

Fig. 4. Visual comparison of classification results (Site 1).

learners (see site 1 top left and right side of site 2). Also, RF tends to confuse grass for agriculture more often, see center block in site 2. Deep learners also avoid classification of bright buildings as bare land, an error present by the RF classifier on the right side of site 2. Agriculture also seems to be captured more consistently by the deep learners, with more consistent shapes when spatial information is incorporated (ST-LSTM and C-LSTM). In general, C-LSTM maps tend to have a less salt-and-pepper effect than the other maps, which could be attributed to the addition of convolutional features that more efficiently incorporate the pixel's neighborhood context.

4.3. *Value of multi-sensor Landsat observations*

Considering the long temporal span of Landsat observations and the multiple sensors involved one important question is whether classification accuracy is *consistent* across Landsat sensors. To investigate this, an additional accuracy assessment was conducted by simulating (not retraining) the pre-trained C-LSTM on validation sequences on various sensor combinations. It should be noted that Landsat 7 sequences included exclusively pixels not affected by the Scan Line Corrector failure. Results are reported in Table 4, with individual sensor annual sequences extracted from the entire and the spatial edge datasets. Annual sequences may vary in space and time depending on data availability, however due to the large number of sequences comparisons are considered valid.

Considering that small accuracy differences are expected due to spatiotemporal sample variability, all three Landsat sensors offer similar classification accuracies and the same result is obtained for spatial edge samples, though with lower accuracies. This is particularly important as it speaks to the legacy of the Landsat program; it offers extensive monitoring capabilities with consistent observations of similar high quality since at least the launch of Landsat 5 in 1985. The slightly better Landsat 8 performance may be due to more precise and better-quality instruments. Landsat 7 minor underperformance may be a result of its longer temporal overlap with our reference dataset leading to higher potential for temporal variability due to environmental dynamics. Landsat sensors comparison reported in other research also shows slightly better performance of Landsat 8 compared to Landsat 5 and Landsat 7 (Poursanidis et al., 2015; Liem et al., 2019) but it may also depend on the classifier type and its parameters (He et al., 2015).

Another important question is whether having multiple *concurrent* Landsat sensors offers classification accuracy benefits. NASA and USGS

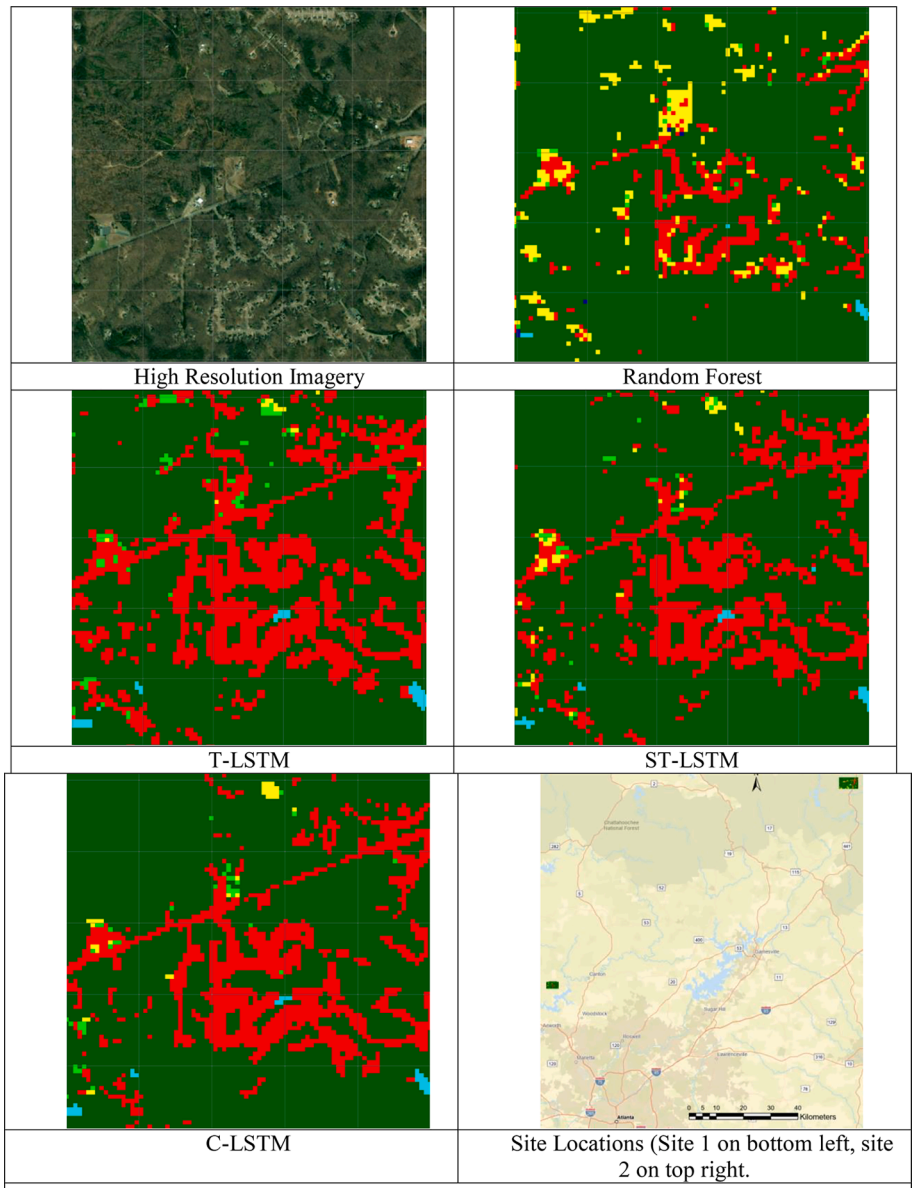

Fig. 5. Visual comparison of classification results (Site 2).

Table 4
Overall and class F1 statistics for single Landsat sensor (sample spatial locations may differ).

Entire Dataset										
Sensor	# sequences	Overall Ac.	Aver. F1	Water	Imp	Grass	For	Bare	Agr	Wetld
Landsat 5	1,240,935	94.2%	94.8%	98.4%	95.1%	91.3%	94.6%	98.0%	93.3%	93.1%
Landsat 7	2,297,126	93.8%	94.1%	98.1%	95.4%	91.2%	93.4%	97.7%	92.5%	90.6%
Landsat 8	882,002	94.9%	95.1%	98.6%	96.3%	91.9%	94.4%	97.0%	93.6%	94.3%
Spatial Edge samples with differing neighborhood = 1										
Sensor	***# sequences***	***Overall Ac.***	***Aver. F1***	***Water***	***Imp***	***Grass***	***For***	***Bare***	***Agr***	***Wetld***
Landsat 5	399,236	87.5%	87.7%	92.9%	90.6%	80.5%	88.6%	89.7%	86.1%	85.3%
Landsat 7	735,631	86.3%	86.1%	92.3%	90.5%	79.0%	86.5%	88.7%	84.2%	81.7%
Landsat 8	281,439	88.0%	88.0%	93.4%	92.0%	81.5%	88.1%	86.8%	87.2%	86.8%

have been advocating for two or more Landsat sensors operating simultaneously as a safeguard for sensor failures but also to study earth dynamics requires shorter revisit times (Wulder et al., 2019; Wu et al., 2019). Here, we investigate the value of two concurrently operating Landsat sensors, the fusion of Landsat 5 and 7, and fusion of Landsat 7 and 8 sensors. Unfortunately, all three sensors did not overlap to study them further.

To extract relevant annual sequences from the validation dataset, first sequences having two sensor information were extracted. Then from those multi-sensor sequences single sensor observations were extracted to create matching sequences in space and time (e.g., a specific pixel in a specific year). This led to two single sensor sequences and one fusion sequence for common spatial locations and years.

The results are shown in Tables 5 and 6, respectively. Comparisons

Table 5
Overall and class F1 statistics for Landsat 5, 7 and their fusion (same sample spatial locations).

Entire Dataset										
Sensor	# sequences	Overall Ac.	Aver. F1	Water	Imp	Grass	For	Bare	Agr	Wetld
Landsat 5	1,240,603	94.2%	94.8%	98.5%	95.1%	91.3%	94.6%	98.0%	93.4%	93.1%
Landsat 7	1,240,603	93.5%	93.9%	97.8%	95.0%	90.9%	93.5%	97.7%	92.3%	89.9%
Landsat 5 + 7	1,240,603	98.0%	98.2%	99.2%	98.1%	97.5%	97.0%	99.4%	98.3%	98.2%
Spatial Edge samples with differing neighborhood = 1										
Sensor	***# sequences***	***Overall Ac.***	***Aver. F1***	***Water***	***Imp***	***Grass***	***For***	***Bare***	***Agr***	***Wetld***
Landsat 5	399,116	87.4%	87.7%	92.9%	90.6%	80.5%	88.6%	89.7%	86.1%	85.3%
Landsat 7	399,116	85.7%	85.6%	91.9%	89.9%	78.6%	86.5%	88.1%	83.7%	80.5%
Landsat 5 + 7	399,116	95.0%	95.1%	96.1%	95.9%	92.8%	93.1%	96.3%	95.6%	95.6%

Table 6
Overall and class F1 statistics for Landsat 7, 8 and their fusion (same sample spatial locations).

Entire Dataset										
Sensor	# sequences	Overall Ac.	Aver. F1	Water	Imp	Grass	For	Bare	Agr	Wetld
Landsat 7	881,781	93.5%	93.7%	98.3%	95.5%	90.5%	92.6%	97.3%	91.5%	90.3%
Landsat 8	881,781	94.9%	95.1%	98.6%	96.3%	91.9%	94.4%	97.0%	93.6%	94.3%
Landsat 7 + 8	881,781	98.1%	98.2%	99.3%	98.3%	97.4%	96.7%	99.4%	98.2%	98.4%
Spatial Edge samples with differing neighborhood = 1										
Sensor	***# sequences***	***Overall Ac.***	***Aver. F1***	***Water***	***Imp***	***Grass***	***For***	***Bare***	***Agr***	***Wetld***
Landsat 7	281,357	85.6%	85.2%	92.3%	90.2%	77.2%	85.4%	88.1%	82.7%	80.8%
Landsat 8	281,357	88.7%	88.0%	93.5%	92.0%	81.5%	88.1%	86.7%	87.2%	86.8%
Landsat 7 + 8	281,357	95.1%	95.1%	96.2%	96.2%	92.6%	92.7%	96.6%	95.6%	95.8%

can be made directly within each table but only indirectly between the two tables as they are composed of different locations/years (although the large number of samples reduces variability). The fusion of Landsat 5/7 or Landsat 7/8 provides a considerable gain, especially in the difficult to classify spatial edge samples where improvements in average F1 accuracy approach 10%. Classes benefiting more from Landsat fusion are the grass (up to 15.4%) and wetland (up to 15.1%), followed by agriculture (up to 12.9%). Results are consistent across 5/7 and 7/8 Landsat fusion. These improvements are substantial when considering available improvement room from single sensor observations and speak to the value of overlapping Landsat sensors, even for high level LCLU classification schemes with basic classes. They are also consistent with prior work by Liem et al. (2019) and Bonansea et al. (2018).

5. Discussion and conclusions

Landsat has been the workhorse of medium resolution environmental analysis. After USGS's decision to make Landsat data freely available, usage has exponentially increased. Data organization and manipulation capabilities offered by cloud platforms such as the Google Earth Engine coupled with advanced parallel computing power make continuous, consistent, near real-time earth monitoring feasible. USGS's Land Change Monitoring, Assessment, and Projection (LCMAP) activities are a prime example of harvesting the extensive Landsat record.

Classification accuracy of large-scale monitoring hovers around 80%, see Grekousis et al. (2015) for a comparison of global mapping products, leaving room for improvement. Deep learning methods have offered significant advancements in other fields of study and as illustrated in our literature review, they have recently received considerable attention for analysis of earth observations. One major limiting factor constraining deep learners from reaching their full potential is the relatively small reference datasets typically employed in our community. As shown by Heydari and Mountrakis (2019) in a *meta*-analysis of deep neural networks in remote sensing these small datasets have led to accuracy saturation. Here, our in-house substantially larger dataset supported for the first time the thorough investigation of the deep learning methods applicability on the extensive Landsat archive.

Results clearly show there are considerable gains compared to traditionally employed classifiers, such as Random Forests. While classes such as water and forest have been found easy to classify by other studies, our results show that improvements can be substantial for traditionally difficult to classify classes such as bare and developed (for example see Wickham et al., 2021, for their assessed performance for NLCD map). These improvements can be attributed to two architectural benefits of deep learning methods. Firstly, borrowing from applications in other fields, they easily support extensive time-series analysis. This is paramount to take advantage of the extensive Landsat temporal record. Traditional classifiers, such as Random Forests or Support Vector Machines require a predefined temporal length (e.g., 4 scenes per year). This is an important limitation as scene availability is not guaranteed, for example due to cloud coverage. Here, our reported results take this limitation into account and approach the RF implementation from the practical implementation perspective. Therefore, the obtained RF accuracies are the combined effect of data and algorithmic limitations. Another issue with RF methods is that they do not architecturally supporting direct temporal linkages between these different timestamps as each input is treated independently. For example, the green band in time t and the green band in time t + 1 are not algorithmically internally linked through a temporal dependency. The second architectural benefit of deep learning methods is the ability to automatically extract spatial features. Typically, spatial features are pre-defined, then calculated and inserted as additional input vectors (e.g., the standard deviation of a 3x3 window on an NDVI layer). Deep learners, and in particular convolutional neural networks, have the ability to automatically extract features of interest as part of their training process. Although spatial relationships are not as pronounced at the 30 m Landsat scale when compared to other high-resolution datasets (e.g., for face recognition), it was demonstrated that they still hold considerable explanatory value.

While simulation times can vary considerably depending on hardware and number of available Landsat scenes, it is helpful to provide demonstrative numbers. We simulated one Landsat scene for the entire year 2016 using all cloud free observations (see Fig. 5). The RF method took 4762 s using CPU resources (Intel Core i7 @ 4.2 GHz), while the T-LSTM, ST-LSTM and C-LSTM took 5622, 10,790 and 16,851 s respectively using GPU resources (NVIDIA RTX A5000 with 16 GB of RAM + AMD Ryzen 9 3950X CPU @ 3.5 GHz). These numbers do not include data preparation, which could also add significantly to processing times. Also, none of the methods were optimized in their implementations. In general, it seems there is a 2x-3x simulation time cost when moving to deep learners, which is not inconsiderable for large scale product

generation.

This work should be considered as the starting point for deep learning methods implementation in large-area monitoring. The architectures employed, the combinations of CNN to capture spatial relationships and LSTM for temporal relationships, are common architectures and the field is constantly advancing. While we optimized our networks for the given architectures, there are numerous and continuous advancements in the deep learning field, that could offer further improvements. Our community has already started to examine different architectures. One example is using another sequence processing cell named GRU (Gated Recurrent Unit) instead of the employed LSTM. The GRU structure is simpler than the LSTM, but has showed good performance in various applications of remote sensing data such as crop disease detection by Bi et al. (2020), hyperspectral image classification by Pan et al. (2020), or combined CNN + GRU architecture for crop classification by Li et al. (2019). Specialized convolutional modules are another path of innovation. Examples include depthwise separable convolution used in Yu et al. (2020) to develop a less complex yet powerful design, integrating a CNN network with attention mechanisms to weight features more efficiently (Tian et al., 2021), or multi-level encoder-decoder approach of fully convolutional network in Nemni et al. (2020).

With respect to sampling, our dataset was intentionally based exclusively on earth observations (Landsat and topography). This supports generalization at multiple spatial and temporal extents. However, the developed models are not yet ready for continental application. Even though no accuracy deficiencies were associated with specific classes, different sampling designs should be considered, especially with respect to rare classes. More importantly, our validation dataset, while not spatially overlapping with the calibration data, was still extracted from adjacent pixels within the 84 blocks. For proper accuracy assessment of a continental classifier, random samples generated across the U.S. would be needed. Finally, our sample dataset only considered Landsat observations from Landsat missions 5, 7, and 8. With the recent launch of Landsat 9 and its expected improvements over prior Landsat missions while keeping its data continuity, more enhanced model performance seems readily achievable (Masek et al., 2020). There are current efforts to harmonize Landsat and Sentinel-2 data which could offer additional monitoring capabilities for a short historical period and moving into the future (see Q. Wang et al., 2017; E. D. Chaves et al., 2020; Shang and Zhu, 2019).

The work presented in this manuscript falls under pixel-based classifiers. There is another approach to classification, a scene-based one. In the latter case, an entire patch is presented to an algorithm and a single label is assigned to it (e.g. airplane, house). Scene-based approaches are more applicable to observations of higher spatial resolution (1–4 m), where individual land objects can be seen and extracted. In the future, it would be interesting to investigate fusion of pixel and scene based methods along with fusion of datasets of different spatial resolution.

To the best of our knowledge, this is the first investigation of deep learning methodologies that uses a substantially large reference dataset while it amplifies algorithmic performance differences through the use of LCLU spatial edge samples for targeted evaluation. The benefits of the deep learning methodologies are evident and are also consistent across all LCLU classes. Our work also examined the practical value of having two, instead of one, Landsat sensors concurrently mapping our planet. Results suggest there are substantial classification increases across all classes through sensor fusion, further justifying the decision to have at least two Landsat sensors in orbit at all. This is important not only for redundancy in case of a sensor malfunction but also provides improved mapping capabilities, even for basic Anderson Level I classification schemes.

Future work could include multiple topics. Firstly, a randomly distributed dataset would be necessary to further train and assess a deep learning method from the operational perspective of continental mapping. Also, repetition of the experiment with different calibration/validation dataset distributions and repetitions would increase confidence in the obtained results. Secondly, additional deep architectures could be tested to identify a good balance between network complexity and mapping accuracy. Thirdly, it is important to consider fusion with Sentinel sensors, as both information content (shorter revisit times) and type (spatial resolution, radar features) may offer additional benefits.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work was supported by the NASA's Land Cover Land Use Change Program (grant # NNX15AD42G) and NASA High-End Computing facility at NASA Ames Research Center, CA. We are grateful for excellent technical and administrative NASA support, in particular the assistance from Blaise Hartman and Samson Cheung among others. We also appreciate the efforts from Kristi Sayler and Mark Drummond (USGS) for providing the Trends data. We thank our reference data generation team of current and former SUNY ESF students: Atef Alla Eddin Amriche, Harrison Goldspiel, Madusha Sammani, Megan Kate Medwid, Rabia Munsaf Khan, and Zhixin Wang.

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.isprsjprs.2023.05.005.

References

Ahmad, J., Farman, H., Jan, Z., 2019. Deep learning methods and applications. In: Khan, M., Jan, B., Farman, H. (Eds.), Deep Learning: Convergence to Big Data Analytics. Springer Singapore, Singapore, pp. 31–42. https://doi.org/10.1007/978-981-13-3459-7_3. SpringerBriefs in Computer Science.

Alhassan, V., Henry, C., Ramanna, S., Storie, C., 2020. A deep learning framework for land-use/land-cover mapping and analysis using multispectral satellite imagery. Neural Comput. & Applic. 32 (12), 8529–8544.

Andrimont, R., Yordanov, M., Martinez-Sanchez, L., et al., 2020. Harmonised LUCAS in-situ land cover and use database for field surveys from 2006 to 2018 in the European Union. Sci Data 7, 352. https://doi.org/10.1038/s41597-020-00675-z.

Bi, L., Guiping, H., Raza, M.M., Kandel, Y., Leandro, L., Mueller, D., 2020. A gated recurrent units (GRU)-based model for early detection of soybean sudden death syndrome through time-series satellite imagery. Remote Sens. (Basel) 12 (21), 3621. https://doi.org/10.3390/rs12213621.

Bonansea, M., Rodriguez, C., Pinotti, L., 2018. Assessing the potential of integrating landsat sensors for estimating chlorophyll-a concentration in a reservoir. Hydrol. Res. 49 (5), 1608–1617. https://doi.org/10.2166/nh.2017.116.

Campos-Taberner, M., García-Haro, F.J., Martínez, B., Izquierdo-Verdiguier, E., Atzberger, C., Camps-Valls, G., Gilabert, M.A., 2020. Understanding deep learning in land use classification based on sentinel-2 time series. Sci. Rep. 10 (1), 17188. https://doi.org/10.1038/s41598-020-74215-5.

Chang, T., Rasmussen, B., Dickson, B., Zachmann, L., 2019. Chimera: a multi-task recurrent convolutional neural network for forest classification and structural estimation. Remote Sens. (Basel) 11 (7), 768. https://doi.org/10.3390/rs11070768.

Chaves, M.E.D., Picoli, M.C.A., Sanches, L.D., 2020. Recent applications of Landsat 8/OLI and Sentinel-2/MSI for land use and land cover mapping: a systematic review. Remote Sens. (Basel) 12 (18), 3062. https://doi.org/10.3390/rs12183062.

Chen, J., Yang, K., Chen, S., Yang, C., Zhang, S., He, L., 2019. Enhanced normalized difference index for impervious surface area estimation at the Plateau Basin Scale. J. Appl. Remote Sens. 13 (01), 19. https://doi.org/10.1117/1.JRS.13.016502.

Chen, B., Zheng, H., Wang, L., Hellwich, O., Chen, C., Yang, L., Liu, T., Luo, G., Bao, A., Chen, X.i., 2022. A joint learning Im-BiLSTM model for incomplete time-series Sentinel-2A data imputation and crop classification. Int. J. Appl. Earth Obs. Geoinf. 108 (April), 102762 https://doi.org/10.1016/j.jag.2022.102762.

Corbane, C., Syrris, V., Sabo, F., Politis, P., Melchiorri, M., Pesaresi, M., Soille, P., Kemper, T., 2021. Convolutional Neural networks for global human settlements mapping from Sentinel-2 Satellite Imagery. Neural Comput. & Applic. 33 (12), 6697–6720. https://doi.org/10.1007/s00521-020-05449-7.

Deng, L., 2014. Deep learning: methods and applications. Foundations and Trends® in Signal Processing 7 (3–4), 197–387. https://doi.org/10.1561/2000000039.

Erb, K.-H., Kastner, T., Plutzar, C., Anna, L.S., Bais, N.C., Fetzel, T., Gingrich, S., et al., 2018. Unexpectedly large impact of forest management and grazing on global

vegetation biomass. Nature 553 (7686), 73–76. https://doi.org/10.1038/nature25138.
Garajeh, K., Mohammad, T.B., Haghi, V.H., Weng, Q., Kamran, K.V., Li, Z., 2022. A Comparison Between Sentinel-2 and Landsat 8 OLI satellite images for soil salinity distribution mapping using a deep learning convolutional neural network. Can. J. Remote. Sens. April, 1–17. https://doi.org/10.1080/07038992.2022.2056435.
Giri, Chandra P., 2016. Remote sensing of land use and land cover: principles and applications.
González-Vélez, J.C., Martinez-Vargas, J.D., Torres-Madronero, M.C., 2022. Land cover classification using CNN and semantic segmentation: a case of study in Antioquia, Colombia. In: Fabián R. Narváez, Julio Proaño, Paulina Morillo, Diego Vallejo, Daniel González Montoya, and Gloria M. Díaz (eds.) Smart Technologies, Systems and Applications. 1532:306–17. Communications in Computer and Information Science. Cham: Springer International Publishing. https://doi.org/10.1007/978-3-030-99170-8_22.
Grekousis, G., Mountrakis, G., Kavouras, M., 2015. An Overview of 21 Global and 43 Regional land-cover mapping products. Int. J. Remote Sens. 36 (21), 5309–5335. https://doi.org/10.1080/01431161.2015.1093195.
Güneralp, B., Zhou, Y., Ürge-Vorsatz, D., Gupta, M., Yu, S., Patel, P.L., Fragkias, M., Li, X., Seto, K.C., 2017. Global scenarios of urban density and its impacts on building energy use through 2050. Proceedings of the National Academy of Sciences 114 (34), 8945–8950. https://doi.org/10.1073/pnas.1606035114.
Hall-Beyer, M., 2017. Practical guidelines for choosing GLCM textures to use in landscape classification tasks over a range of moderate spatial scales. Int. J. Remote Sens. 38 (5), 1312–1338. https://doi.org/10.1080/01431161.2016.1278314.
Hall-Beyer, M., 2017a. GLCM Texture: A Tutorial v. 3.0 March 2017, March. https://doi.org/10.11575/PRISM/33280.
He, J., Harris, J.R., Sawada, M., Behnia, P., 2015. A comparison of classification algorithms using Landsat-7 and Landsat-8 Data for mapping lithology in Canada's Arctic. Int. J. Remote Sens. 36 (8), 2252–2276. https://doi.org/10.1080/01431161.2015.1035410.
Helber, P., Bischke, B., Dengel, A., Borth, D., 2019. EuroSAT: a novel dataset and deep learning benchmark for land use and land cover classification. IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. 12 (7), 2217–2226. https://doi.org/10.1109/JSTARS.2019.2918242.
Heydari, S.S., Mountrakis, G., 2018. Effect of classifier selection, reference sample size, reference class distribution and scene heterogeneity in per-pixel classification accuracy using 26 landsat sites. Remote Sens. Environ. 204 (January), 648–658. https://doi.org/10.1016/j.rse.2017.09.035.
Heydari, S.S., Mountrakis, G., 2019. Meta-analysis of deep neural networks in remote sensing: a comparative study of mono-temporal classification to support vector machines. ISPRS J. Photogramm. Remote Sens. 152 (June), 192–210. https://doi.org/10.1016/j.isprsjprs.2019.04.016.
Interdonato, R., Ienco, D., Gaetano, R., Ose, K., 2018. DuPLO: A DUal View Point Deep Learning Architecture for Time Series ClassificatiOn. ArXiv:1809.07589 [Cs], September. http://arxiv.org/abs/1809.07589.
Iyer, M.S., Rhinehart, R.R., 1999. A method to determine the required number of neural-network training repetitions. IEEE Trans. Neural Netw. 10 (2), 427–432. https://doi.org/10.1109/72.750573.
Jamali, A., Mahdianpari, M., 2022. Swin transformer and deep convolutional neural networks for coastal wetland classification using Sentinel-1, Sentinel-2, and LiDAR Data. Remote Sens. (Basel) 14 (2), 359. https://doi.org/10.3390/rs14020359.
Karra, K., Kontgis, C., Statman-Weil, Z., Mazzariello, J.C., Mathis, M., Brumby, S.P., 2021. Global land use / land cover with Sentinel 2 and Deep Learning. In: 2021 IEEE International Geoscience and Remote Sensing Symposium IGARSS, 4704–7. Brussels, Belgium: IEEE. https://doi.org/10.1109/IGARSS47720.2021.9553499.
Li, Z., Chen, G., Zhang, T., 2019. Temporal attention networks for multitemporal multisensor crop classification. IEEE Access 7, 134677–134690. https://doi.org/10.1109/ACCESS.2019.2939152.
Lin, Z., Zhong, R., Xiong, X., Guo, C., Jinfan, X.u., Zhu, Y., Jialu, X.u., et al., 2022. Large-scale rice mapping using multi-task spatiotemporal deep learning and Sentinel-1 SAR Time Series. Remote Sens. (Basel) 14 (3), 699. https://doi.org/10.3390/rs14030699.
Liu, L., Zhang, X., Gao, Y., Chen, X., Shuai, X., Mi, J., 2021. Finer-resolution mapping of global land cover: recent developments, consistency analysis, and prospects. Journal of Remote Sensing 2021 (March), 1–38. https://doi.org/10.34133/2021/5289697.
Lyu, H., Hui, L.u., Mou, L., 2016. Learning a transferable change rule from a recurrent neural network for land cover change detection. Remote Sens. (Basel) 8 (6), 506. https://doi.org/10.3390/rs8060506.
Ma, L., Liu, Y.u., Zhang, X., Ye, Y., Yin, G., Johnson, B.A., 2019. Deep learning in remote sensing applications: a meta-analysis and review. ISPRS J. Photogramm. Remote Sens. 152 (June), 166–177. https://doi.org/10.1016/j.isprsjprs.2019.04.015.
Masek, J.G., Wulder, M.A., Markham, B., McCorkel, J., Crawford, C.J., Storey, J., Jenstrom, D.T., 2020. Landsat 9: empowering open science and applications through continuity. Remote Sens. Environ. 248 (October), 111968 https://doi.org/10.1016/j.rse.2020.111968.
Masolele, R.N., De Sy, V., Herold, M., Marcos, D., Verbesselt, J., Gieseke, F., Mullissa, A. G., Martius, C., 2021. Spatial and temporal deep learning methods for deriving land-use following deforestation: a pan-tropical case study using landsat time series. Remote Sens. Environ. 264 (October), 112600 https://doi.org/10.1016/j.rse.2021.112600.
Mazzia, V., Khaliq, A., Chiaberge, M., 2019. Improvement in land cover and crop classification based on temporal features learning from sentinel-2 data using recurrent-convolutional neural network (R-CNN). Appl. Sci. 10 (1), 238. https://doi.org/10.3390/app10010238.
Nemni, E., Bullock, J., Belabbes, S., Bromley, L., 2020. Fully convolutional neural network for rapid flood segmentation in synthetic aperture radar imagery. Remote Sens. (Basel) 12 (16), 2532. https://doi.org/10.3390/rs12162532.
Ojala, T., Pietikainen, M., Maenpaa, T., 2002. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. IEEE Trans. Pattern Anal. Mach. Intell. 24 (7), 971–987. https://doi.org/10.1109/TPAMI.2002.1017623.
Pan, E., Mei, X., Wang, Q., Ma, Y., Ma, J., 2020. Spectral-spatial classification for hyperspectral image based on a Single GRU. Neurocomputing 387 (April), 150–160. https://doi.org/10.1016/j.neucom.2020.01.029.
Pelletier, C., Webb, G., Petitjean, F., 2019. Temporal convolutional neural network for the classification of satellite image time series. Remote Sens. (Basel) 11 (5), 523. https://doi.org/10.3390/rs11050523.
Pengra, B.W., Stehman, S.V., Horton, J.A., Dockter, D.J., Schroeder, T.A., Yang, Z., Hernandez, A.J., Healey, S.P., Cohen, W.B., Finco, M.V., Gay, C., Houseman, I.W., 2020, LCMAP Reference Data Product 1984-2018 land cover, land use and change process attributes (ver. 1.2, November 2021): U.S. Geological Survey data release, https://doi.org/10.5066/P9ZWOXJ7.
Pérez-Hoyos, A., Rembold, F., Kerdiles, H., Gallego, J., 2017. Comparison of global land cover datasets for cropland monitoring. Remote Sens. (Basel) 9 (11), 1118. https://doi.org/10.3390/rs9111118.
Pimm, S.L., Jenkins, C.N., Abell, R., Brooks, T.M., Gittleman, J.L., Joppa, L.N., Raven, P. H., Roberts, C.M., Sexton, J.O., 2014. The biodiversity of species and their rates of extinction, distribution, and protection. Science 344 (6187), 1246752. https://doi.org/10.1126/science.1246752.
Pongratz, J., Schwingshackl, C., Bultan, S., Obermeier, W., Havermann, F., Guo, S., 2021. Land use effects on climate: current state, recent progress, and emerging topics. Current Climate Change Reports 7 (4), 99–120. https://doi.org/10.1007/s40641-021-00178-y.
Poursanidis, D., Chrysoulakis, N., Mitraka, Z., 2015. Landsat 8 vs. Landsat 5: a comparison based on urban and peri-urban land cover mapping. Int. J. Appl. Earth Obs. Geoinf. 35 (March), 259–269. https://doi.org/10.1016/j.jag.2014.09.010.
Rousset, G., Despinoy, M., Schindler, K., Mangeas, M., 2021. Assessment of deep learning techniques for land use land cover classification in Southern New Caledonia. Remote Sens. (Basel) 13 (12), 2257. https://doi.org/10.3390/rs13122257.
Rußwurm, M., Körner, M., 2017. Multi-temporal land cover classification with long short-term memory neural networks. ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences XLII-1/W1 (May): 551–58. https://doi.org/10.5194/isprs-archives-XLII-1-W1-551-2017.
Rußwurm, M., Körner, M., 2018. Multi-temporal land cover classification with sequential recurrent encoders. ArXiv:1802.02080 [Cs], February. http://arxiv.org/abs/1802.02080.
Saadeldin, M., O'Hara, R., Zimmermann, J., Namee, B.M., Green, S., 2022. Using deep learning to classify grassland management intensity in ground-level photographs for more automated production of satellite land use maps. Remote Sens. Appl.: Soc. Environ. 26 (April), 100741 https://doi.org/10.1016/j.rsase.2022.100741.
Shang, R., Zhu, Z., 2019. Harmonizing Landsat 8 and Sentinel-2: a time-series-based reflectance adjustment approach. Remote Sens. Environ. 235 (December), 111439 https://doi.org/10.1016/j.rse.2019.111439.
Sharma, A., Liu, X., Yang, X., 2018. Land cover classification from multi-temporal, multi-spectral remotely sensed imagery using patch-based recurrent neural networks. Neural Netw. 105 (September), 346–355. https://doi.org/10.1016/j.neunet.2018.05.019.
Sumbul, A.d.W., Kreuziger, T., Marcelino, F., Costa, H., Benevides, P., Caetano, M., Demir, B., Markl, V., 2021. BigEarthNet-MM: A large scale multi-modal multi-label benchmark archive for remote sensing image classification and retrieval. IEEE Geosci. Remote Sens. Mag. https://doi.org/10.1109/MGRS.2021.3089174.
Sun, Z., Di, L., Fang, H., 2019. Using long short-term memory recurrent neural network in land cover classification on landsat and cropland data layer time series. Int. J. Remote Sens. 40 (2), 593–614. https://doi.org/10.1080/01431161.2018.1516313.
Thorp, K.R., Drajat, D., 2021. Deep machine learning with sentinel satellite data to map paddy rice production stages across West Java, Indonesia. Remote Sens. Environ. 265 (November), 112679 https://doi.org/10.1016/j.rse.2021.112679.
Tian, T., Li, L., Chen, W., Zhou, H., 2021. SEMSDNet: a multiscale dense network with attention for remote sensing scene classification. IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. 14, 5501–5514. https://doi.org/10.1109/JSTARS.2021.3074508.
Van Liem, N., Van Bao, D., Bac, D.K., Hieu, N., Hieu, D.T., Van Phong, T., Ha, T.T.V., Nga, P.T.P., Trinh, P.T., 2019. Integrating Landsat 7 and 8 data to improve basalt formation classification: a case study at Buon Ma Thuot Region, Central Highland, Vietnam. Open Geosciences 11 (1), 901–917. https://doi.org/10.1515/geo-2019-0070.
Wang, Q., Blackburn, G.A., Onojeghuo, A.O., Dash, J., Zhou, L., Zhang, Y., Atkinson, P. M., 2017b. Fusion of Landsat 8 OLI and Sentinel-2 MSI Data. IEEE Trans. Geosci. Remote Sens. 55 (7), 3885–3899. https://doi.org/10.1109/TGRS.2017.2683444.
Wang, J., X. Li, S. Zhou, Tang, J., 2017. Landcover classification using deep fully convolutional neural networks. In: AGU Fall Meeting Abstracts, 2017:IN11E-02.
Wang, Y., Zhang, Z., Feng, L., Ma, Y., Qingyun, D.u., 2021. A new attention-based CNN approach for crop mapping using time series Sentinel-2 Images. Comput. Electron. Agric. 184 (May), 106090 https://doi.org/10.1016/j.compag.2021.106090.
Wickham, J., Stehman, S.V., Sorenson, D.G., Gass, L., Dewitz, J.A., 2021. Thematic accuracy assessment of the NLCD 2016 Land Cover for the Conterminous United States. Remote Sens. Environ. 257 (May), 112357 https://doi.org/10.1016/j.rse.2021.112357.
Wu, Z., Snyder, G., Vadnais, C., Arora, R., Babcock, M., Stensaas, G., Doucette, P., Newman, T., 2019. User needs for Future Landsat Missions. Remote Sens. Environ. 231 (September), 111214 https://doi.org/10.1016/j.rse.2019.111214.

Wulder, M.A., Loveland, T.R., Roy, D.P., Crawford, C.J., Masek, J.G., Woodcock, C.E., Allen, R.G., et al., 2019. Current status of Landsat Program, Science, and Applications. Remote Sens. Environ. 225 (May), 127–147. https://doi.org/10.1016/j.rse.2019.02.015.

Yu, D., Qing, X.u., Guo, H., Zhao, C., Lin, Y., Li, D., 2020. An efficient and lightweight convolutional neural network for remote sensing image scene classification. Sensors 20 (7), 1999. https://doi.org/10.3390/s20071999.

Zhang, X., Ling, D., Tan, S., Fangming, W., Zhu, L., Zeng, Y., Bingfang, W.u., 2021. Land use and land cover mapping using RapidEye imagery based on a novel band attention deep learning method in the Three Gorges Reservoir Area. Remote Sens. (Basel) 13 (6), 1225. https://doi.org/10.3390/rs13061225.

Zhang, L., Zhang, L., Bo, D., 2016. Deep learning for remote sensing data: a technical tutorial on the state of the art. IEEE Geosci. Remote Sens. Mag. 4 (2), 22–40. https://doi.org/10.1109/MGRS.2016.2540798.

Zhao, X., Gao, L., Chen, Z., Zhang, B., Liao, W., 2019. Large-scale landsat image classification based on deep learning methods. APSIPA Transactions on Signal and Information Processing 8, e26.

Zhao, B., Huang, B.o., Zhong, Y., 2017. Transfer learning with fully pretrained deep convolution networks for land-use classification. IEEE Geosci. Remote Sens. Lett. 14 (9), 1436–1440. https://doi.org/10.1109/LGRS.2017.2691013.

Zhu, M., He, Y., He, Q., 2019. A review of researches on deep learning in remote sensing application. Int. J. Geosci. 10 (01), 1–11. https://doi.org/10.4236/ijg.2019.101001.