*remote sensing*

MDPI

*Article*

# High-Quality Cloud Masking of Landsat 8 Imagery Using Convolutional Neural Networks

**M. Joseph Hughes * and Robert Kennedy**

College of Earth, Ocean, and Atmospheric Science, Oregon State University, Corvallis, OR 97331, USA; rkennedy@coas.oregonstate.edu

* Correspondence: m.joseph.hughes@gmail.com

Received: 25 August 2019; Accepted: 1 November 2019; Published: 5 November 2019


**Abstract:** The Landsat record represents an amazing resource for discovering land-cover changes and monitoring the Earth's surface. However, making the most use of the available data, especially for automated applications ingesting thousands of images without human intervention, requires a robust screening of cloud and cloud-shadow, which contaminate clear views of the land surface. We constructed a deep convolutional neural network (CNN) model to semantically segment Landsat 8 images into regions labeled clear-sky, clouds, cloud-shadow, water, and snow/ice. For training, we constructed a global, hand-labeled dataset of Landsat 8 imagery; this labor-intensive process resulted in the uniquely high-quality dataset needed for the creation of a high-quality model. The CNN model achieves results on par with the ability of human interpreters, with a total accuracy of 97.1%, omitting only 3.5% of cloud pixels and 4.8% of cloud shadow pixels, which is seven to eight times fewer missed pixels than the masks distributed with the imagery. By harnessing the power of advanced tensor processing units, the classification of full images is I/O bound, making this approach a feasible method to generate masks for the entire Landsat 8 archive.

**Keywords:** Landsat; cloud masking; cloud-shadow; convolutional neural network; image segmentation; deep learning

## 1. Introduction

The sensors aboard Landsat 8 have been collecting high-quality imagery of the Earth since 2013. Free and open to the public, with global wall-to-wall coverage of land surfaces at an ecologically meaningful spatial resolution of 30 m, Landsat imagery is one of the most useful resources for ecological monitoring and wildland management [1,2]. However, harnessing the power of the Landsat archive to detect and describe changes on the Earth's surface hinges on researchers' ability to detect and aggregate clear-sky observations uncontaminated by clouds and cloud-shadow [3–6].

Due to this necessity, screening for clouds has always been essential for making the full use of Landsat's spectral imagery. Early algorithms were designed to provide scene-level estimates to be included in metadata and enable human operators to make informed decisions before purchasing and downloading imagery [7]. Later, more robust algorithms attempted to generate masks of clear-sky views within scenes and used ancillary information in addition to the spectral values in the image, such as elevation, sun angle, and cloud temperature [8] or the spatial relationships of predictions within a scene [9]. Currently, Landsat 8 imagery is accompanied by a bitwise quality mask (BQA) that encodes information about each pixel's quality and includes masks for clouds, cloud-shadows, and snow/ice. These are generated using the CFMask algorithm, which was shown to be both reliable over a large evaluation dataset as well as computationally suitable for application over the entire archive [10].

In the last decade, deep convolutional neural networks (CNN) have revolutionized image recognition [11]. Deep CNNs are fundamentally neural networks with two key modifications.

The 'deep' designation means that instead of having only one or a few hidden layers, they have dozens, enabling complex features to be constructed from more primitive features learned at early layers of the network. 'Convolutional' refers to the network learning sets of two-dimensional convolutional filters that are applied across the image, as opposed to a traditional neural network that treats each input pixel of an image independently. Learning small filters greatly decreases the number of total weights to learn while also allowing flexibility as to where in the image objects are located. Early versions of these algorithms generated either a single conceptual label or a list of labels with associated probabilities for the subject of the scene [12–15]. These networks have since been used across a variety of remote sensors to classify land covers or create cloud masks by feeding the networks chips from the image and, typically, predicting a single central pixel [16–20].

Recently, CNNs have been applied to the cloud and cloud-shadow detection problem, including state-of-the-art algorithms to detect atmospheric obstructions in Meteosat [21], Sentinel-2 [22], and multiple sensors [23,24]. Additionally, several deep learning approaches have been developed for Landsat imagery [25–27], in part due to free access to the high-quality training and evaluation data for these sensors that was used to validate the CFMask algorithm, and which is freely available from the United State Geological Survey. These data includes the Landsat 7 and Landsat 8 Biome Cloud Cover Assessment Validation Data (Biome) as well as the dataset developed for training and evaluating the algorithm described in this paper (SPARCS, or Spatial Procedures for Automated Removal of Cloud and Shadow) [10]. Some of these algorithms also use semantic segmentation approaches and represent important improvements with classification accuracy rates of approximately 91% on the SPARCS dataset [28], which approaches human accuracy.

Two insights make training semantic segmentation with CNNs functional. First, deconvolution layers enable the network to produce outputs in a higher resolution than the inputs; these are applied after pooling several times to re-expand the receptive field back into the original resolution. The convolution plus pooling ensures that the network learns spatially relevant information about the image set, and the deconvolution allows the network to meaningfully apply that information when determining relationships between nearby pixels [29]. Second, if all layers in the network are convolutional (i.e., a fully convolutional network, FCN), then the input size of the image to be classified is constrained only by hardware, rather than being fixed to the size chosen during training [30]. In the cases where a typical CNN architecture with an intermediate dense, fully connected layer predicts a central region of the input image, that central region size is fixed, and the strided chips of the original image need to be fed into the network and then reassembled. In the most extreme cases, only a single central pixel is predicted from the strided chip. Many of the convolutions from the border around that central region, which inform the central classifications, could also be used to predict pixels neighboring that central region, but are instead discarded between strided predictions. This wastes significant computation, since many of the same convolutions are performed multiple times on the same data. In a FCN, the size of the central region is not fixed (though the 'border' size is), so one can simply supply more of the original image and receive a larger area of predictions, reducing repeated computation.

Technological advances in the form of graphical processing units specially designed for training neural networks, dubbed tensor processing units (TPUs), enable comparatively rapid training and evaluation of complex network architectures. TPUs reduce training time from weeks (typical when running on CPUs) to hours and image prediction from hours to the few seconds needed to read and write the data with negligible processing time [31].

Humans can easily identify clouds and cloud shadows in most single-date Landsat imagery when given spatial context. This insight led to the SPARCS algorithm (Spatial Procedures for Automated Removal of Cloud and Shadow), which was developed in 2014 for the Landsat 4 and 5 Thematic Mapper [9], and here we extend it to Landsat 8. Similar to the original SPARCS, we take a neural network approach, although here we use a many-layer network instead of the original single-layer network. Further diverging, the "Spatial Procedures" are built into the network in the form of

convolutional and deconvolutional layers, which the network uses to learn which spatial features are important to the cloud and cloud-shadow identification task.

In this manuscript we present a fully automated algorithm capable of identifying clouds and cloud-shadows in Landsat 8 imagery that produces errors on par with humans while also being sufficiently computationally efficient to feasibly process the entire archive. We describe the neural network architecture of our method and compare the method to the quality bands distributed with Collection 1 Landsat data and an additional third-party dataset of imagery.

## 2. Materials and Methods

### *2.1. Training and Evaluation Data*

Landsat imagery is captured and organized geographically along the WRS2 path/row system. Eighty unique path/rows were selected by stratifying across each of the 14 World Wildlife Fund terrestrial Major Habitat Types (MHTs) plus 'Inland Water' and 'Rock and Ice' for each of the seven biogeographical realms (64 scenes, not all habitat types occur in each realm), plus an additional scene from each MHT chosen randomly [32] (Figure 1). For each path/row, a single Landsat 8 image acquired during 2013 through 2014 was selected at random and downloaded from the USGS Earth Explorer. Since data from different areas within a single Landsat image have similar land cover and share atmospheric conditions and acquisition variables such as sun angle, manually classifying an entire scene is a redundant use of classification effort. Instead, a single 1000 px × 1000 px subscene was selected non-randomly from each image to ensure the presence of the habitat type of interest and, where possible, a mix of clouds, clear-sky, and water.

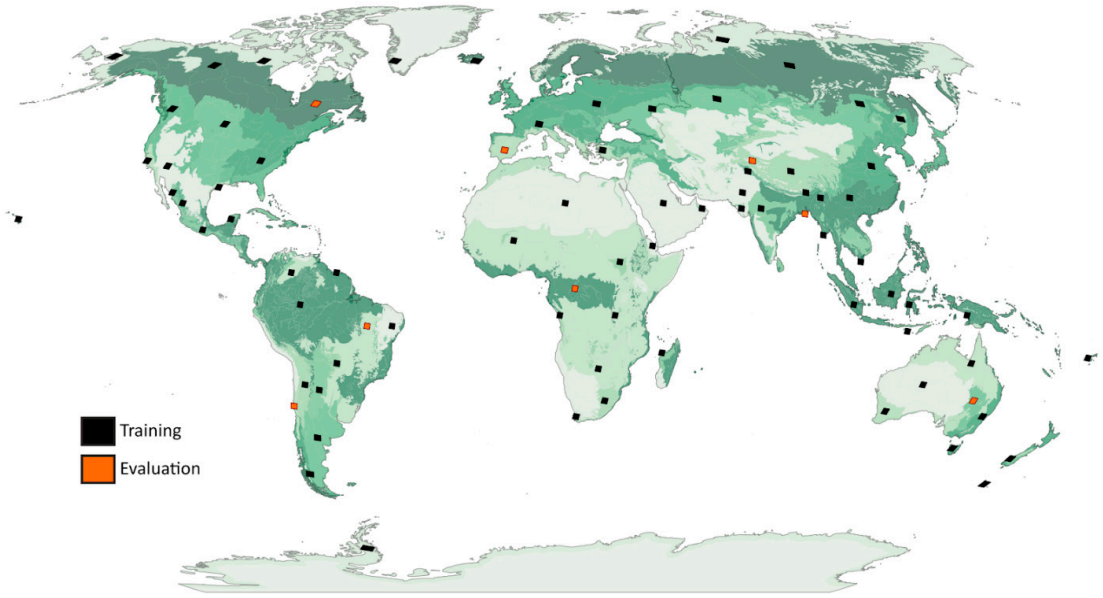


**Figure 1.** Locations were selected by identifying one path/row from each of the 14 World Wildlife Fund terrestrial Major Habitat Types plus 'Inland Water' and 'Rock and Ice' (these 16 types represented in shades of green) for each of the seven biogeographical realms, plus an additional scene from each habitat type chosen at random. Not all combinations occur; a total of 80 scenes were selected and split into 72 used during training (black) and eight used during evaluation (orange).

To facilitate manual labeling, false-color images were generated by mapping the shortwave-infrared-1 band (B6) to red, the near-infrared band (B5) to green, and the red band (B4) to blue. A single interpreter labeled each pixel based only on visual interpretation using Photoshop; no thresholding, clustering, or other automated/mathematical approaches were used to assist labeling. This is in contrast to other publically available datasets, such as the Landsat 8 Biome Cloud Cover

Assessment Validation Data (Biome) [10]. Generating the training data in this manner removes many telltale artefacts in label masks—areas that are a speckled mixture of two classes, halos between objects and backgrounds arising from gradients, and small or thin objects omitted due to minimum mapping units—and enables more powerful learning algorithms.

Pixels were labeled as either no-data, clear-sky, cloud, cloud-shadow, shadow-over-water, snow/ice, water, or flood. During training and validation, the shadow-over-water class was combined with the shadow class. The flood class was re-coded as water or clear-sky, as appropriate for land-cover type, due to insufficient examples and high spectral variability. Water and snow/ice are both land-cover types and the result of short-term weather conditions. Including them in masks enables analysts to decide how to treat these conditions for a specific problem while also providing additional information to time-series algorithms in support of automated decision making.

This dataset (Dataset S1) is available at http://emapr.ceoas.oregonstate.edu/sparcs/.

### *2.2. Neural Network Architechture*

The machine learning classifier used for the cloud screening task is a deep, fully convolutional neural network with 20.4 million weights (Figure 2). It predicts six classes of interest: no-data, clear-sky, cloud, cloud-shadow, snow/ice, and water. The classifier can be conceptually separated into three phases: 1. convolution, 2. deconvolution, and 3. output.

The convolution phase is characterized by a series of two-dimensional (2D) convolution layers interrupted by $2 \times 2$ max pooling layers. Each 2D convolution layer learns N filters of size $3 \times 3 \times$ Depth, with N (and thereby the depth of the following convolution) increasing as the max pooling layers decrease the effective resolution. The max pooling layers examine each non-overlapping $2 \times 2$ window and pass through only the maximum value, reducing resolution. Through this process, spatial information is aggregated across the image, trading resolution for an increasing number of filters describing increasingly complex spatial relationships. The CNN architecture in this stage contains approximately 16.5 million weights, out of approximately 20.5 million weights for the entire network, and is identical to that in VGG-16 [12]. To benefit from transfer learning, weights in analogous layers are initialized to those from VGG-16 and are prevented from updating for the first 10 epochs to encourage later layers to converge toward using the VGG-16 outputs.

In the deconvolution phase, the information encoding spatial structure is used to reconstruct the spatial resolution. This phase makes use of 2D deconvolution layers, also referred to as the transpose of 2D convolution. These layers learn filters of size learn $2 \times 2 \times$ Depth that each have four separate outputs arranged as a $2 \times 2$ window. In this way, the output from each deconvolution layer doubles the resolution of the input. As the resolution increases, the number of filters learned and used for prediction decreases, in an inverse pattern to the convolution phase. After two of these upscaling steps, the moderate resolution features from Phase 1 are directly added to the deconvolution outputs, allowing the network to use both the spatial information reconstructed from the large-scale features and the more moderate-scale features. The deconvolution phase fully returns the data back to its original resolution.

The output phase contains a novel feature of our network: data flow splits into two branches to discourage the network from simply using fine-scaled features. Both branches predict the same labels, but in the first, no fine-scale spatial features are included, forcing the network to learn useful features for the classification task in the convolution and deconvolution phases. In the second, the early fine-scale features are combined with the output from the deconvolution phase to enable the network to fine-tune spatial structure. The loss from these outputs is combined, with the loss from the first weighted twice as strongly as the loss from the second. Only the second branch, which includes the fine-scale features, is used during prediction. Additionally, 2D spatial dropout [33] is used as a regularization layer. During training, this sets 3/8 of the features to 0, forcing the network to learn redundant patterns that ideally correspond with different avenues of evidence. During prediction, the dropout is omitted to allow all features to contribute to classification.

Due to memory constraints during training, a 28-pixel border is clipped from all sides of the output within the network. This could have been resolved by using a smaller window size during training, but since the edges of each input image incorporate many no-data pixels, clipping has the benefit of removing the least informed predictions.

All layers prior to the final prediction layer use a reticulated unit activation function; the final layer uses a softmax activation to convert activation energies to class probabilities.

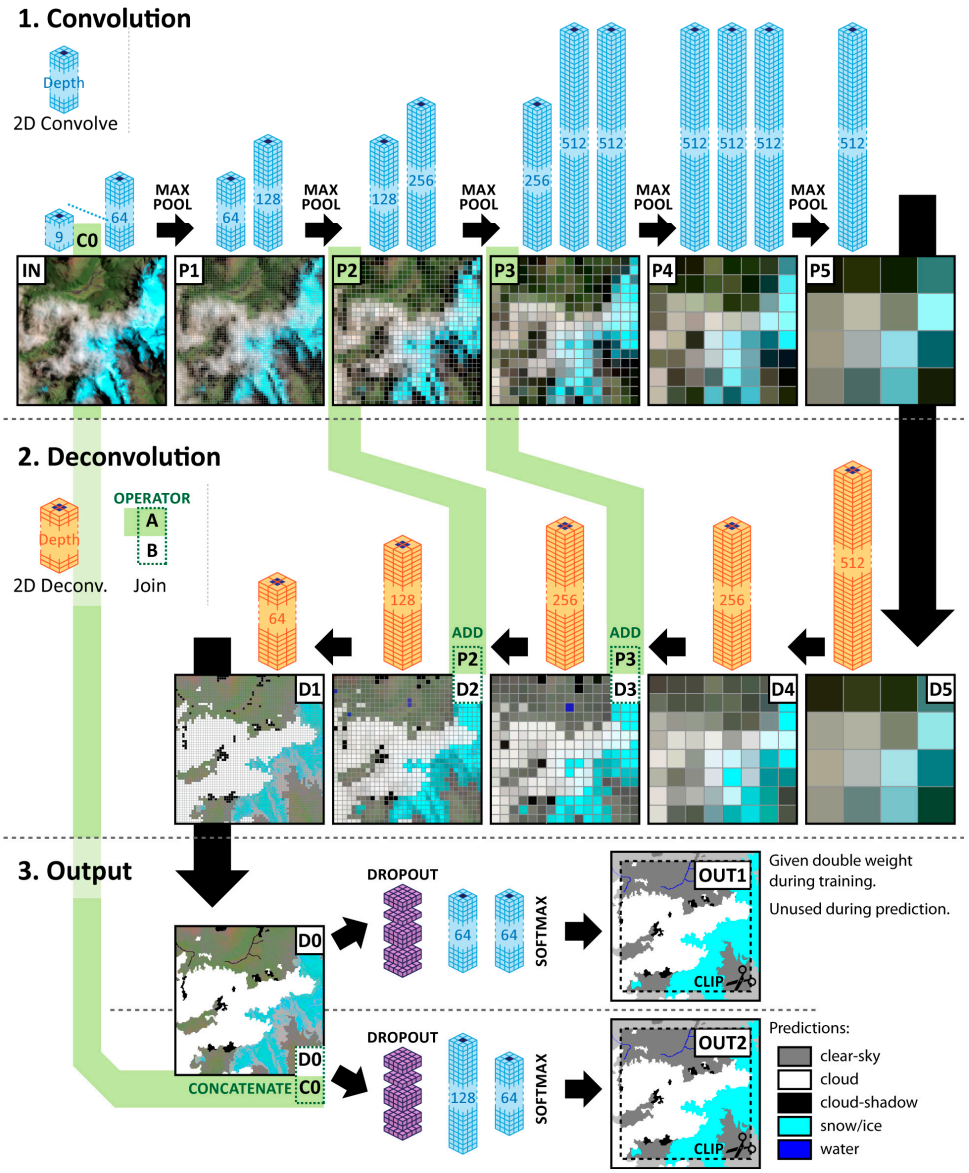


**Figure 2.** Convolutional neural network (CNN) architecture for the cloud and cloud-shadow screening task. In phase one, increasingly complex spatial features are extracted by a series of two-dimensional convolution layers (blue) and max pooling. Blue numbers within convolutional blocks denote the number of features each convolutional filter uses. For each labeled image (IN, P1-P5), the resolution is half of the one preceding it. In the second phase, these features are used to reconstruct the spatial resolution through a series of deconvolution layers (orange), with selected earlier layers (P2, P3) summed into outputs to contribute earlier detail (green). In this phase, the resolution of images re-doubles between numbered images (D5–D0). In the final phase, two outputs are predicted, one from only the deconvolution output (OUT1) and one that also combines fine-scale features from early in the network to ensure fine-scale predictions (OUT2); only the second output is used during prediction. Outputs are clipped to remove a 28 pixel border from all sides.

### 2.3. Processing

Prior to processing, reflectance values in Landsat 8 imagery were corrected to top of atmosphere reflectance [34]. All spectral bands were used except for the 15 m panchromatic band (B8). The two thermal bands were summed to create a single thermal feature to avoid any spurious information arising from varied processing around the stray light issue and to facilitate adapting the network to other Landsat sensors, resulting in a total of nine predictive features. Each feature was normalized using the feature-wise mean and standard deviation of the training dataset.

Six subscenes were set aside for validation and two were set aside for testing, leaving 72 for training. These 1000 px × 1000 px subscenes were padded to include a 64 px border of no-data to simulate predicting at the edges of the Landsat scenes. During training, each requested example was a 256 px × 256 px window randomly clipped from one of these padded subscenes. Each epoch consists of 1440 samples from all subscenes, and the network was allowed to proceed for up to 100 epochs or until the validation sample stopped improving for five epochs, whichever occurred first. In practice, all epochs ended early due to a lack of validation improvement. Training examples were batched and randomized following best practices [35]. The network predicts a central 200 px × 200 px region from the 256 px × 256 px window during training.

Each training example presented to the network is equivalent to 40,000 single-pixel examples. Since these are all contiguous, stratified sampling based on class is not possible. To partially mitigate this issue, weighted Kullback–Leibler divergence [36] was used as the loss function, with pixels labeled clear-sky given half weight. Clear-sky pixels make up approximately 65% of the total dataset, but are also the most spectrally diverse class. This simple reduction was sufficient to allow the network to reliably discern less frequent classes.

The model was fit using the Adam optimizer with default parameters [37].

During prediction, the network produces classifications for a region equal to the input size minus a 28-px buffer along each edge. For this paper, the full 1000 px × 1000 px validation and testing subscenes were no-data buffered by 28 pixels and predicted in a single pass through the network.

Training and evaluation were performed using TensorFlow in Python, with the CNN model specified with Keras. Computation was performed using Google Cloud Services and the TensorFlow Research Cloud. Data was stored in the TFRecord format in Google Cloud Bucket, program control was executed using Google Compute Engine, and training was accelerated using tensor processing units (TPUs). TFRecord is the format recommended by the TensorFlow Data Input Pipeline best practices manual [35], and provides a way to compactly serialize and store information for efficient retrieval across the Google Compute Engine network. This stack decreased computation time by several orders of magnitude compared to a multi-processor CPU scenario, allowing rapid exploration and the experimentation of different architectures and hyperparameters.

### 2.4. Evaluation

The CNN model was evaluated against the two test scenes as well as the six validation scenes. The test scenes were not used during training, whereas validation scenes were used to monitor progress and determine early stopping. Cohen's kappa [38], full confusion matrices, and accuracy and recall metrics are calculated between the predicted labels and the manual labels for each subscene. Then, these results were compared to those calculated between the quality bands included by USGS with the Landsat data and the manual labels. For the quality band masks, a pixel defaulted to clear-sky but was considered cloud if the cloud flag was set (bit 4), a cloud-shadow if the high cloud-shadow bit was set (bit 8), and snow/ice if the high snow/ice bit was set (10). For cloud-shadow and snow/ice, this corresponds to labeling pixels when the CFMask algorithm has medium or high confidence in the condition. CFMask does not distinguish water; these pixels were considered clear-sky during evaluation.

Since clouds and cloud-shadows have fuzzy boundaries, we allowed two pixels of leeway at cloud and cloud-shadow borders within the manual labeled masks, where either of the classes at the boundary were counted as correct. This was used for masks generated with both SPARCS and CFMask,

and assures that reported errors are not simply confusing small amounts at edges but are real failures to detect objects or the full extent of objects.

Finally, four subscenes were selected to be manually labeled twice to measure interpreter consistency and the limit of human accuracy. These subscenes were reflected and rotated and then reinterpreted a year after the initial interpretation. The results between these interpretations are combined and presented as a single confusion matrix.

## 3. Results

### *3.1. Performance of CNN SPARCS*

The results from the two test scenes (path/row (PR) 201/033 and PR 148/035) were within the range of the results from the six validation scenes, and so the eight were combined for analysis. Confusions are combined into a single matrix for all evaluation scenes for presentation.

The new CNN version of SPARCS performs very well, with a total accuracy of 97.1% (Table 1) and a Cohen's kappa of 0.947 over the eight subscenes. The recall percentages for each class fall between 92% and 98%, meaning that users can trust masks resulting from the method to be consistent. In comparison, the quality bands (Table 2) achieve a total accuracy of 90.9% and a Cohen's kappa of 0.796. The quality bands perform worst for cloud-shadow, finding only 60.4% of cloud-shadows while also only correctly labeling shadows 69.9% of the time. For automated methods, detecting clouds and cloud-shadows in order to mask out those pixels is the most important task, and the CNN SPARCS algorithm omits only 3.5% of cloud pixels and 4.8% of cloud-shadow pixels. This is nearly an order of magnitude improvement over CFMask, which omitted 23.8% of clouded pixels and 39.8% of cloud-shadow pixels.

**Table 1.** Agreement between the CNN Spatial Procedures for Automated Removal of Cloud and Shadow (SPARCS) method described in this paper and the manually labeled imagery across all eight evaluation scenes, with class-wise accuracy and recall statistics (italics).

| | **Clear-Sky** | **Cloud** | **Shadow** | **Snow/Ice** | **Water** | ***Recall*** |
|---|---|---|---|---|---|---|
| **Clear-Sky** | 5,185,970 | 27,372 | 18,209 | 35,057 | 15,755 | *98.2%* |
| **Cloud** | 37,807 | 1,004,243 | 3399 | 2052 | 1563 | *95.7%* |
| **Shadow** | 26,711 | 5993 | 494,661 | 1541 | 10,199 | *91.8%* |
| **Snow/Ice** | 14,509 | 1837 | 1973 | 407,209 | 212 | *95.6%* |
| **Water** | 20,419 | 2057 | 3154 | 4229 | 673,863 | *95.8%* |
| *Accuracy* | *98.1%* | *96.4%* | *94.9%* | *90.5%* | *96.0%* | *97.1%* |

**Table 2.** Agreement between CFMask quality masks and the manually labeled imagery across all eight evaluation scenes, with class-wise accuracy and recall statistics (italics). The 'water' class is not distinguished by CFMask and is included with clear-sky.

| **CFMask** | **Clear-Sky** | **Cloud** | **Shadow** | **Snow/Ice** | ***Recall*** |
|---|---|---|---|---|---|
| **Clear-Sky** | 5,874,317 | 218,065 | 204,209 | 19,264 | *93.0%* |
| **Cloud** | 27,099 | 793,830 | 693 | 114,182 | *84.8%* |
| **Shadow** | 85,715 | 18,543 | 313,738 | 31,022 | *69.9%* |
| **Snow/Ice** | 195 | 365 | 1143 | 285,620 | *99.4%* |
| *Accuracy* | *98.1%* | *77.0%* | *60.4%* | *63.5%* | *90.9%* |

The spatial relationship of errors varies between the CNN SPARCS and CFMask results, which can be seen in the results from the two test scenes (Figures 3 and 4). For each scene, the false color image and the manual labels are presented along with the predicted results from SPARCS and CFMask, with the differences (minus the two pixel buffer) highlighted below the results from each algorithm. Water is combined with clear-sky for CFMask.

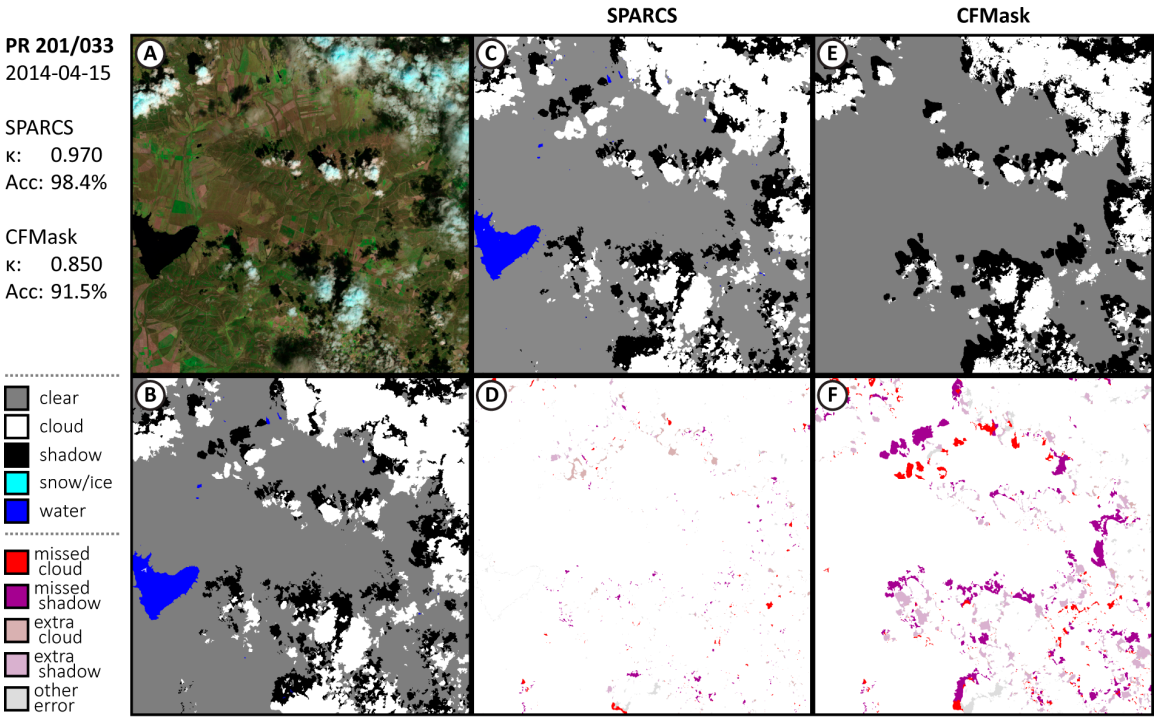


**Figure 3.** Results over test image at path/row 201/033, showing false color image used during interpretation (**A**), the manually labeled interpretations (**B**), with generated masks from SPARCS (**C**) and CFMask (E) with respective spatial distribution of errors (**D**,**F**).

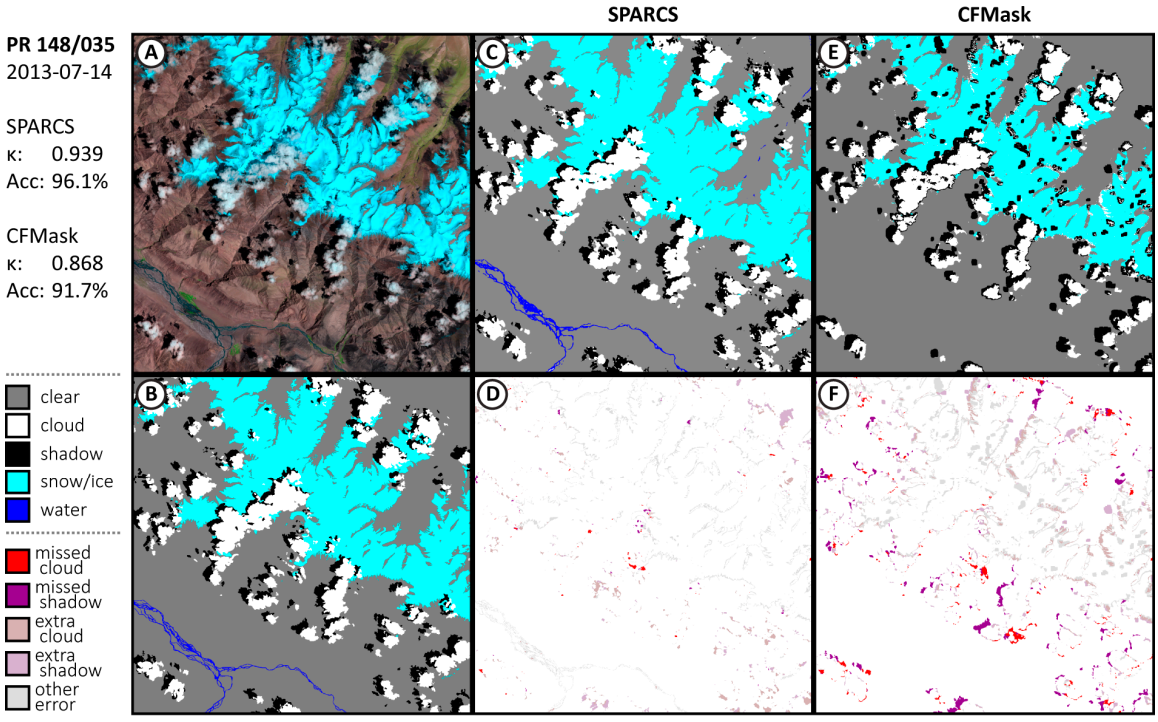


**Figure 4.** Results over test images at path/row 148/035 showing false color image used during interpreation (**A**), the manually labeled interpretation (**B**), with generated masks from SPARCS (**C**) and CFMask (E) with respective spatial distribution of errors (**D**,**F**).

For the CFMask masks, most of the errors come from missing clouds and shadows, which are shown as red and purple in the error images. This is strongly in evidence in the image from PR 201/033 (Figure 3). These types of errors are most important for automated methods because they contaminate

the data stack used for multi-temporal analysis or building mosaics. CFMask especially misses parts of cloud-shadows, due to attempting to predict their position based on properties of the cloud object rather than relying on the spatial information in the image. The error in the SPARCS masks is more balanced among confusion between classes. Most missed cloud and cloud-shadow occurs as a few extra pixels around detected objects, and can be mitigated by dilating the masks. A substantial error occurs in the PR 148/035 image (Figure 4), where the algorithm erroneously identifies a terrain shadow near a cloud as a cloud-shadow in the northeastern corner of the image.

Results for the six validation scenes can be found in Appendix A (Figures A1–A6).

The algorithm was further evaluated using 24 cloud and cloud-shadow masks from the Landsat 8 Biome Cloud Cover Assessment Validation Data (Biome) [10], which was used in evaluation of the CFMask algorithm. Masks were generated using the CNN SPARCS algorithm and compared to Collection 1 BQAs generated by CFMask for the same scenes (masks provided with the Biome dataset are pre-Collection 1). These 24 scenes were selected from the 32 validation scenes that included cloud-shadow and represent a range of land-cover types. Since the CNN SPARCS algorithm also predicts water and snow/ice in addition to clouds, cloud-shadow, and clear-sky, predictions of water and snow/ice were counted as 'clear-sky' for this comparison, as the Biome masks do not include these classes. The same two-pixel allowance around objects used in other comparisons was also used here.

Results for both the CNN SPARCS method and CFMask over the Biome dataset are comparable to the results using our own dataset. The SPARCS algorithm achieved 96% accuracy over all 24 scenes, with a 6% omission error and a 9% commission error. In comparison, CFMask produced 91% accuracy with a 14% omission error and a 14% commission. Appendix A contains afull table of results for each scene (Table A1) along with representative graphical examples (Figures A7–A10).

### *3.2. Human Interpreter Consistency*

Since clouds and cloud-shadows have indeterminate boundaries, some degree of subjectivity in manual labeling is expected. To explore reasonable upper bounds for accuracy results, four images were manually labeled twice by the same interpreter, one year apart. These scenes were selected to provide a range of cloud and land-cover types, and do not represent a statistical sample. The interpreter agreed with himself approximately 96% of the time (Table 3).

**Table 3.** Self-agreement across four images manually labeled twice by the same interpreter, one year apart, with class-wise agreement rates (italics).

| | **Clear-Sky** | **Cloud** | **Shadow** | **Snow/Ice** | **Water** | |
|---|---|---|---|---|---|---|
| **Clear-Sky** | 2,573,774 | 22,919 | 19,882 | 9655 | 8456 | *97.7%* |
| **Cloud** | 22,506 | 605,888 | 2289 | 36,063 | 2943 | *90.5%* |
| **Shadow** | 675 | 7 | 240,210 | 4 | 417 | *99.5%* |
| **Snow/Ice** | 5583 | 911 | 47 | 124,801 | 3 | *95.0%* |
| **Water** | 30,341 | 107 | 501 | 1783 | 290,235 | *89.9%* |
| | *97.8%* | *96.2%* | *91.4%* | *72.4%* | *96.1%* | *95.9%* |

As expected, the spatial distribution of disagreement is primarily at the edges of objects and land covers, such as the boundary of the water in the PR 183/064 scene, although there are some cases where dark land cover near cloud-shadow is included in the shadow in PR 201/033 (Figure 5). Much of the disagreement, though, stems from a single scene with a large, diffuse cloud where the interpreter delineated its shadow quite differently, highlighting the ambiguity of images with thin clouds and haze.

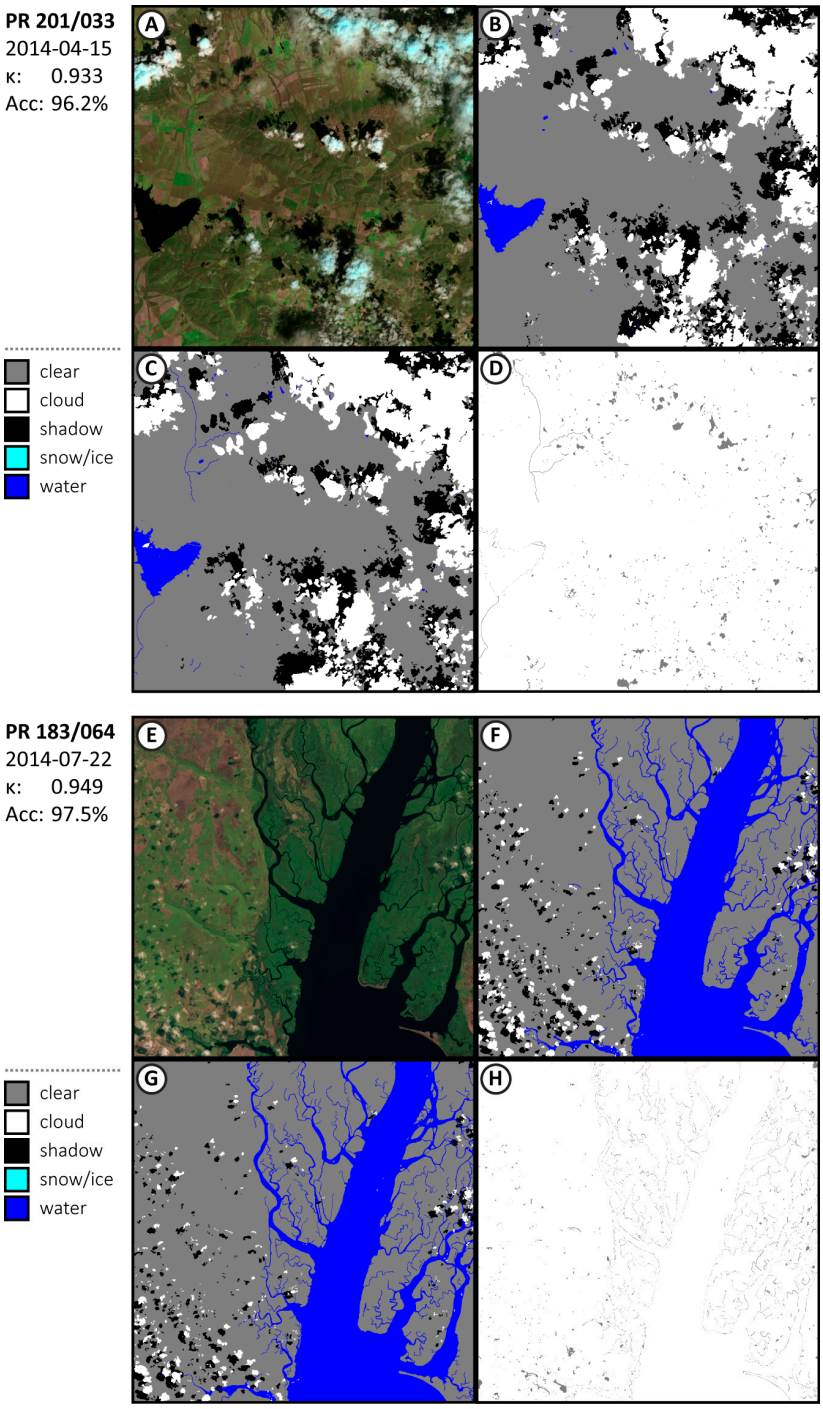


**Figure 5.** Interpreter self-disagreement for two images from PR 201/033 (**top**) and PR 183/064 (**bottom**). Both images were manually labeled by the same interpreter twice, each one year apart. Within each inset: false color image (**A**,**E**), both interpretations (**B**,**C**,**F**,**G**), and the areas of disagreement highlighted in grey (**D**,**H**).

## 4. Discussion

The neural network architecture used here has several design features useful for pixel-wise image segmentation. First, it uses only convolutional operators; therefore, at no time is the 2D spatial arrangement of the input image lost or constrained to be a specific size. This allows for images of any size greater than the reduction by down-sampling (here, 32) to be used during prediction. This provides

flexibility and convenience, as the same network can be used for whole Landsat scenes, small regions of interest, or arbitrarily sized tiles.

Second, the network is able to provide labels for a large number of pixels per pass—omitting only a small border where estimates lose reliability due to edge effects. Compared to CNN methods that provide only a single central output, this method greatly reduces the number of redundant computations, since many of the same convolutions over the same data are needed when estimating neighboring pixels.

This network is large, with 20.5 million weights; so many weights introduces the risk of overfitting the network. Here, we attempt to mitigate that risk using transfer learning. The convolutional phase of the network is initialized with VGG-16, and the remainder of the network is coerced into using those filters by not allowing those weights to be updated in the first few epochs. Such a large network is necessary to provide a large receptive field—i.e., the area around each pixel that is able to inform its classification. One solution to reducing network size while preserving the receptive field is to use convolutions with larger strides in early stages to quickly decrease resolution and then omit later convolution layers with hundreds of layers. However, this also reduces the number of features used to include spatial structure. Dilated convolutions [39] can enlarge the receptive field using fewer layers while retaining feature density. This is a strategy used successfully on similar classification problems discriminating only clouds and cloud-shadows from clear-sky pixels [24,26]. However, early trials in this study with dilated convolutions produced unacceptable outputs with a structured speckle pattern when discriminating between water and shadows from both terrain and clouds; future work may be able to overcome this issue.

For machine learning algorithms to achieve exceptional accuracy, the training data used must itself be of exceptional accuracy. However, the cloud and cloud-shadow classification task has an innate subjectivity given that clouds and their shadows have diffuse edges. Our assessment of this subjectivity found that there was actually more disagreement between images reinterpreted a year apart than disagreement between the CNN classifier and the evaluation scenes. Some of the disagreement in reinterpreted images comes from choosing images with a representative range of land-cover and atmospheric conditions rather than being a statistically representative sample. However, this does not fully explain the image from path/row 201/033, which was used in both the reinterpreted and the evaluation sets. On this image, the algorithm achieved a 98.4% accuracy whereas the reinterpretation only agreed over 96.2% of the image. This is surprising and reflects how the algorithm learned the consistent set of subjective decisions made when the interpreter labeled all of the training imagery, which was performed within a few weeks. However, those judgement calls made by the interpreter shifted to different preferences after a year. Due to this inconsistency, we believe that the CNN algorithm is at or very near the quality that can be performed by human interpreters.

Two challenges hamper the evaluation of cloud and cloud-shadow detection methods. First, accuracy metrics are conceptually skewed—a method that performs at 85% accuracy naively sounds good, but produces useless masks. Second, most users are interested in aggressively removing obstructions; those performing automated time-series analyses are advised to dilate the cloud and cloud-shadow masks to both ensure the entire fuzzy object is covered and to remove the image corruption from the thin haze around such objects. This makes errors at boundaries less important than errors where algorithms miss whole clouds or incorrectly label clear-sky regions as clouds. In the first, dilation is of no help if there is no seed to dilate, and in the second, dilation will subsequently remove large amounts of useable imagery, which could be quite valuable in areas with persistent cloud cover such as the Amazon. Clever methods that perform a size-weighted object detection are needed for proper evaluation.

## 5. Conclusions

A deep convolutional neural network was trained from a global dataset of hand-labeled Landsat 8 imagery to identify regions of clear-sky, clouds, cloud-shadow, snow/ice, and water. The algorithm is

able to perform at the same level as a human interpreter; further increases in accuracy will require either exceedingly careful human interpretation or advances in machine learning. This algorithm generates masks with substantially less error than the masks distributed with the Landsat data, omitting just 3.5% of cloud pixels and 4.8% of cloud-shadow pixels, compared to 23.8% and 39.8%, respectively. By leveraging machine learning techniques to predict large numbers of pixels within images in a single algorithm pass and modern computational hardware in form of tensor processing units, processing the entire Landsat 8 archive is a feasible task. These low error rates over the entire archive will improve the accuracy of—and in some cases, enable—a wide range of algorithms that operate over continental and global scales without requiring human intervention to screen images for cloud and cloud-shadow contamination.

**Supplementary Materials:** The following are available online at http://emapr.ceoas.oregonstate.edu/sparcs/, Dataset S1: Landsat 8 imagery with manually interpreted labels used for training and evaluating the algorithm described here.

**Author Contributions:** Conceptualization, M.J.H.; methodology, M.J.H.; software, M.J.H.; validation, M.J.H.; formal analysis, M.J.H.; investigation, M.J.H.; resources, M.J.H., R.K.; data curation, M.J.H.; writing—original draft preparation, M.J.H.; writing—review and editing, M.J.H., R.K.; visualization, M.J.H.; supervision, M.J.H, R.K.; project administration, M.J.H., R.K; funding acquisition, M.J.H., R.K.

**Funding:** This research was partially funded by the NASA Carbon Monitoring System.

**Acknowledgments:** This research was supported with Cloud TPUs from Google's TensorFlow Research Cloud (TFRC). The authors would like to think the editor and four anonymous reviewers for valuable comments on earlier versions of this manuscript.

**Conflicts of Interest:** The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

## Appendix A

### *Appendix A.1. Additional Results from Validation Subscenes*

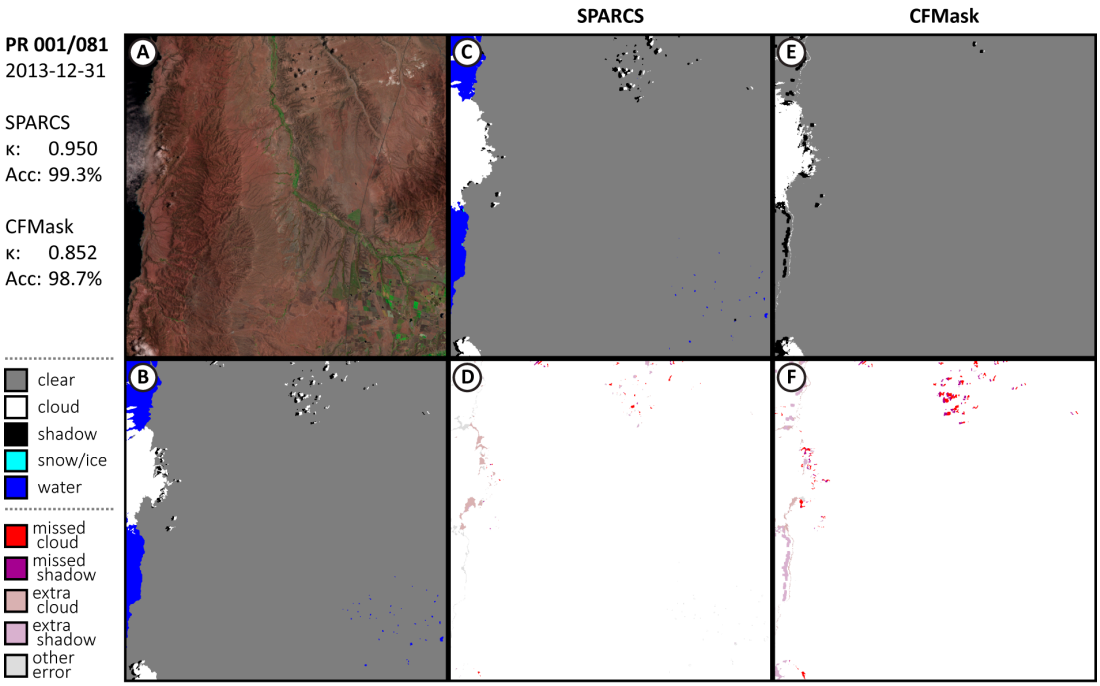


**Figure A1.** Results for validation test image PR 001/081, with false color image used during interpretation (**A**), the manually labeled interpretation (**B**), generated masks from SPARCS (**C**) and CFMask (**E**), and respective spatial distribution of errors (**D**,**F**).

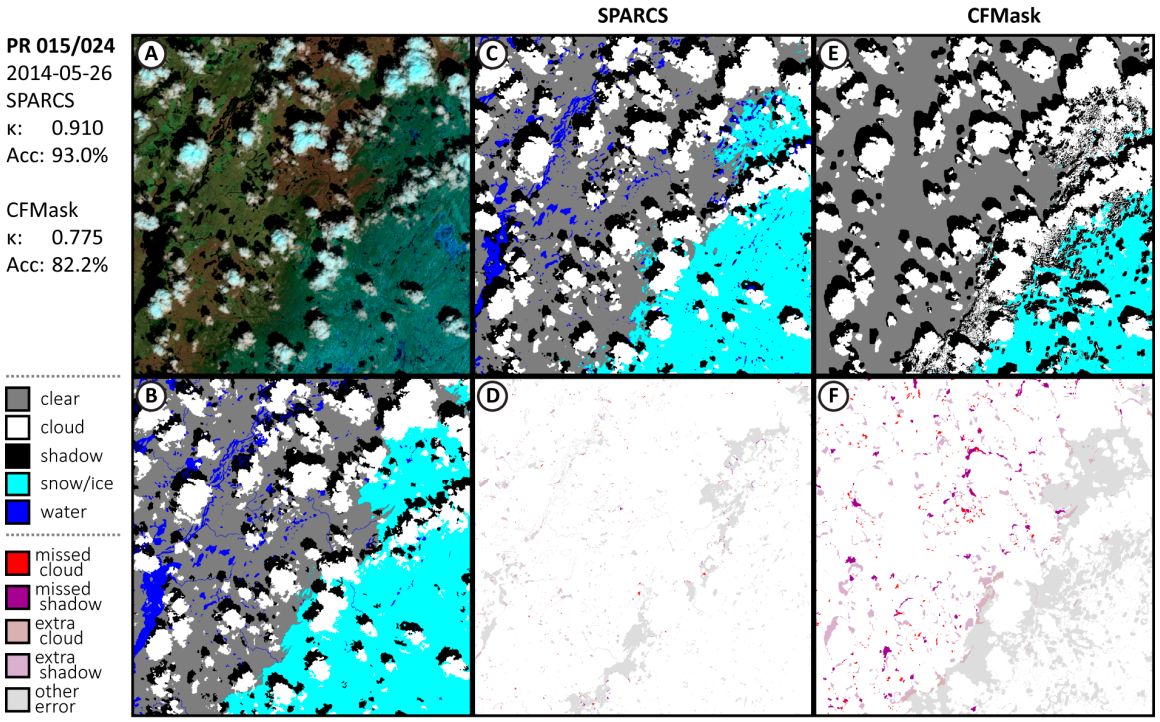


**Figure A2.** Results for validation test image PR 015/024, with false color image used during interpretation (**A**), the manually labeled interpretation (**B**), generated masks from SPARCS (**C**) and CFMask (**E**), and respective spatial distribution of errors (**D**,**F**).

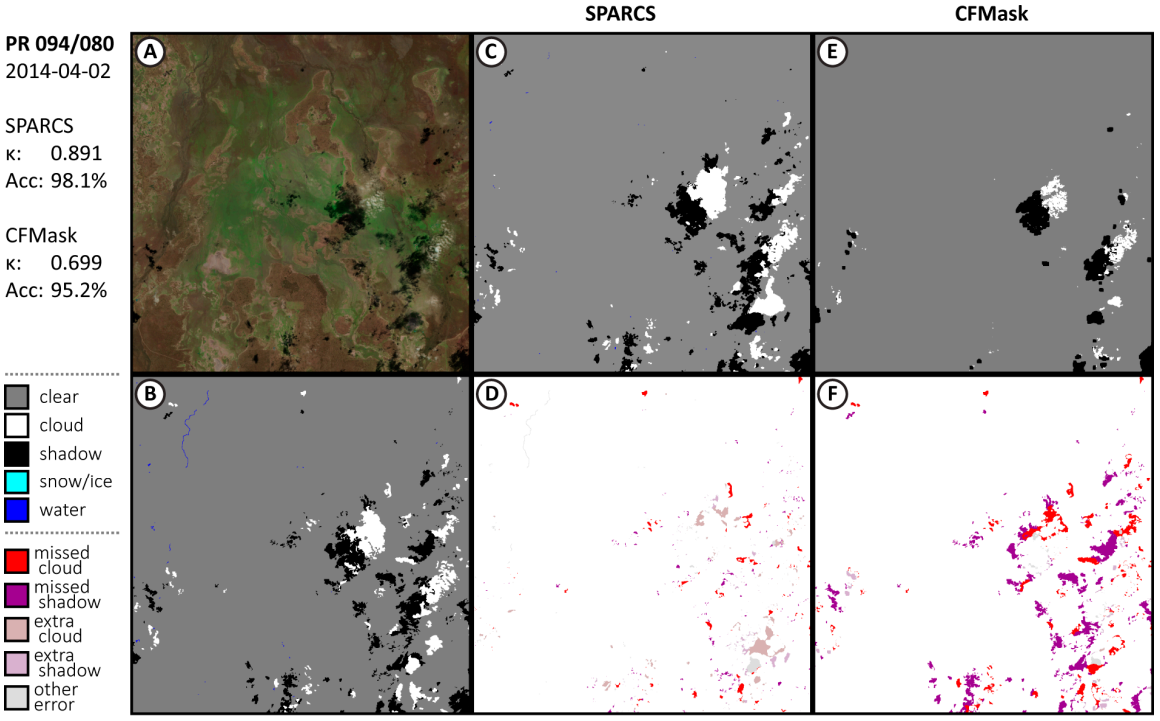


**Figure A3.** Results for validation test image PR 094/080, withfalse color image used during interpretation (**A**), the manually labeled interpretation (**B**), generated masks from SPARCS (**C**) and CFMask (**E**), and respective spatial distribution of errors (**D**,**F**).

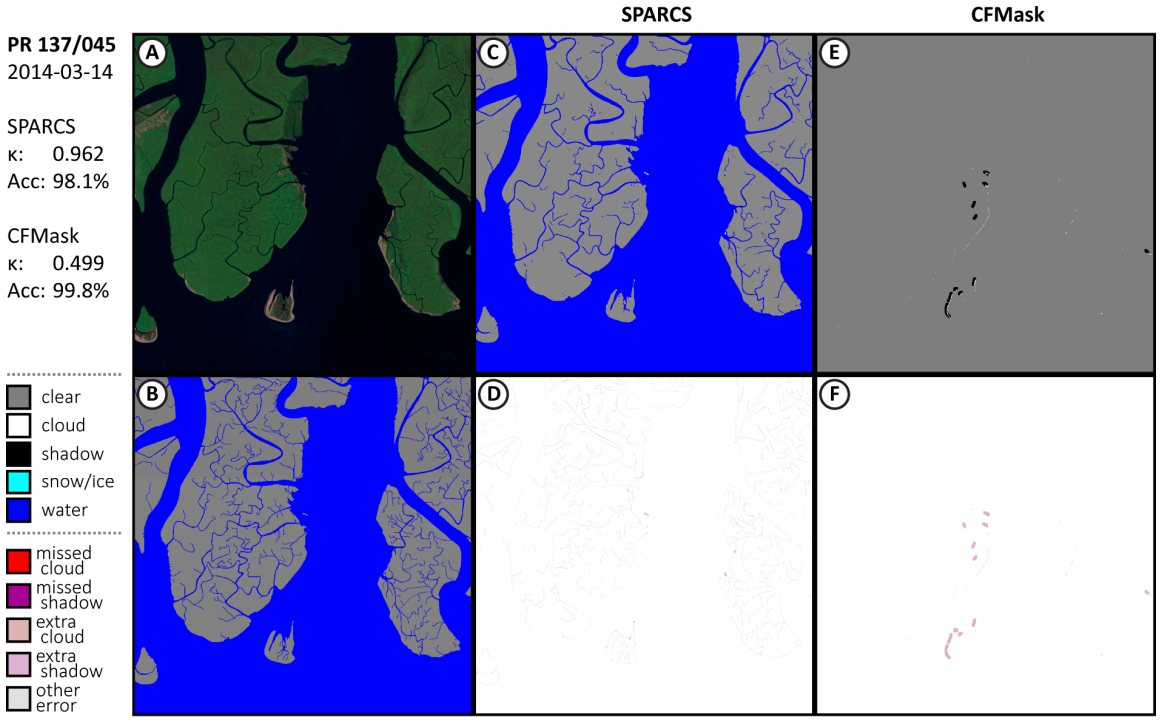


**Figure A4.** Results for validation test image PR 137/045, with false color image used during interpretation (**A**), the manually labeled interpretation (**B**), generated masks from SPARCS (**C**) and CFMask (**E**), and respective spatial distribution of errors (**D**,**F**).

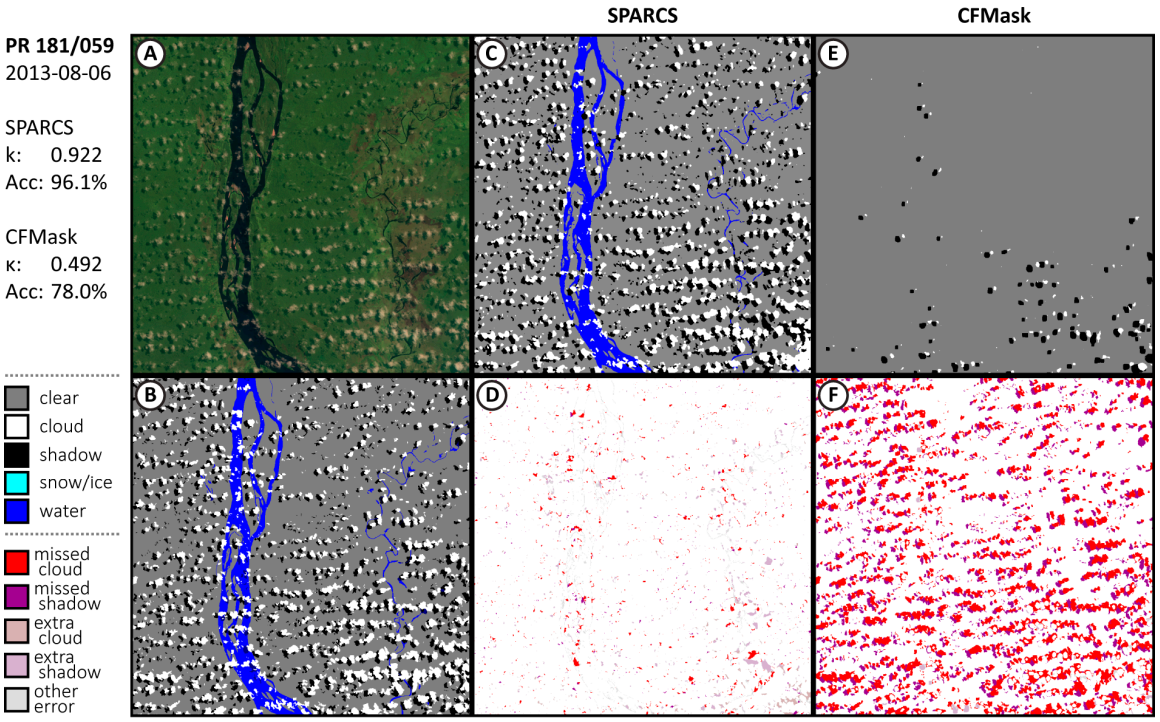


**Figure A5.** Results for validation test image PR 181/059, with false color image used during interpretation (**A**), the manually labeled interpretation (**B**), generated masks from SPARCS (**C**) and CFMask (**E**), and respective spatial distribution of errors (**D**,**F**).

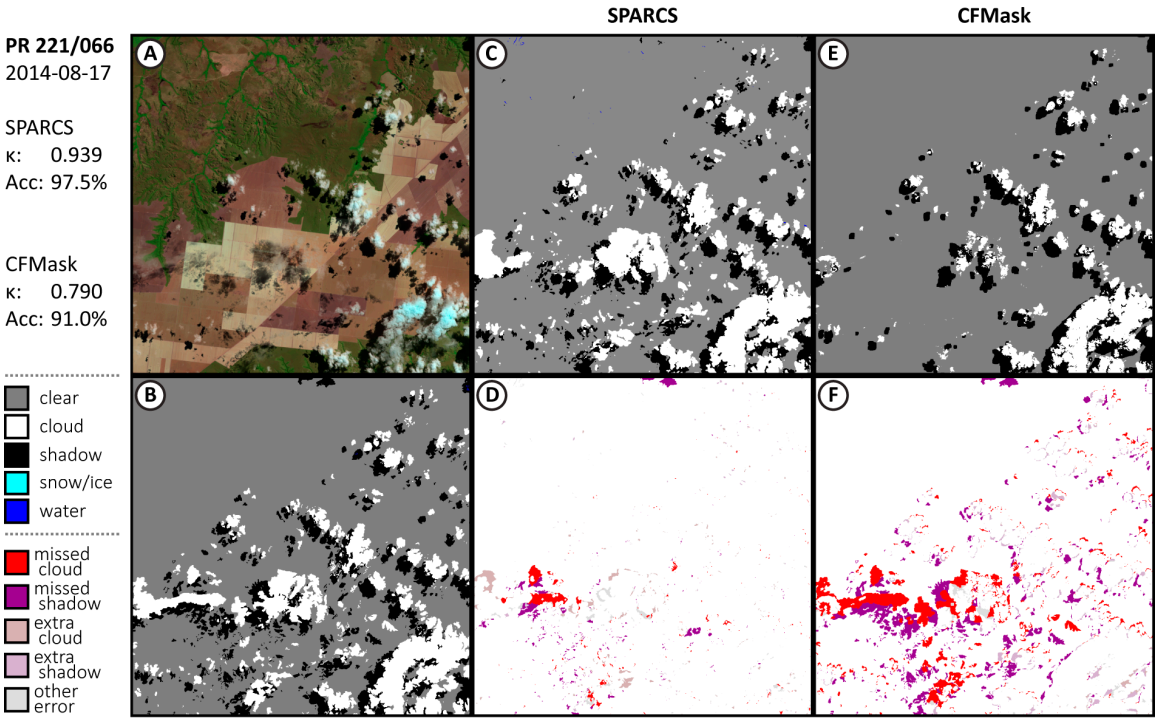


**Figure A6.** Results for validation test image PR 221/066, with false color image used during interpretation (**A**), the manually labeled interpretation (**B**), generated masks from SPARCS (**C**) and CFMask (**E**), and respective spatial distribution of errors (**D**,**F**).

*Appendix A.2. Comparison with Biome Dataset*

Cloud and cloud-shadow masks for 24 additional Landsat 8 scenes from the Landsat 8 Biome Cloud Cover Assessment Validation Data (Biome), which was used in evaluation of the CFMask algorithm, were generated using the CNN SPARCS algorithm and compared to Collection 1 BQAs generated by CFMask for the same scenes (Table A1). The Biome data does not include water or snow/ice; predictions of these classes were counted as clear-sky. Additionally, cloud-shadow is only present in some images and may not have complete coverage even in those images. Only scenes with some cloud-shadow labels were selected. The same two-pixel allowance around objects used in other comparisons was also used here. The CNN SPARCS algorithm achieved 96% accuracy over all 24 scenes, with a 6% omission error and 9% commission error. In comparison, CFMask produced 91% accuracy with 14% omission error and 14% commission. On all but two of the Biome scenes, the CNN SPARCS algorithm produced a higher accuracy than CFMask. In addition, results without that 2-px allowance are also included to facilitate comparisons between algorithms. Results between CNN SPARCS and CFMask are similar for this test, each performing approximately 4% worse in overall accuracy.

Representative 1000 × 1000 px excerpts of Biome scenes are included for comparison (Figures A7–A10). The cropping is performed to enable details to be seen in manuscript figures. In Figure A7, clouds and shadow are over a glacier, emphasizing the continued difficulty in distinguishing clouds and shadows from ice and snow-packed terrain. In Figure A8, a mixture of thin and thick clouds covers the landscape; most of the CNN SPARCS errors are around object edges. Figure A9 includes many small cloud objects; again, most of the disagreement is around object edges, which the human interpreter included in the mask liberally. Figure A10 contains clouds over water; note that the Biome masks do not include shadows over water due to the difficulty in distinguishing between the dark water and the dark shadows. Both CNN SPARCS and CFMask miss some thin clouds over the water in the bottom of the image.

Other deep learning networks have been tested against the Biome dataset and achieve overall accuracies of 94% [27], 95% (Li) [24], and 96.5% [26]. Each of these algorithms is focused on the clouds and cloud-shadow identification problem and do not produce classifications for water or snow/ice. Additionally, they each train on a subset of the Biome data and are evaluated on a different subset and thus are able to internalize any systematic biases, which are more likely to be prominent given that the Biome data is not generated in a fully manual way. The dataset developed for this study has been available from the USGS since 2017. Another study [28] that trained a classifier using the SPARCS data and evaluated it using the Biome dataset provides a better comparison, although again not using the water and snow/ice classes. That classifier performed at 91% accuracy, which is quite similar to the results reported here.

**Table A1.** Kappa scores and overall accuracy (Acc) for SPARCS and the CFMask over 24 Landsat 8 images from the Landsat 8 Biome Cloud Cover Assessment Validation Dataset (Biome).

| | **With 2-px Buffer** | | | | **Without 2-px Buffer** | | | |
|---|---|---|---|---|---|---|---|---|
| | **SPARCS** | | **CFMask** | | **SPARCS** | | **CFMask** | |
| **Scene Identifier** | **Kappa** | **Acc.** | **Kappa** | **Acc.** | **Kappa** | **Acc.** | **Kappa** | **Acc.** |
| LC80010732013109LGN00 | 0.813 | 94.2% | 0.712 | 88.3% | 0.763 | 92.1% | 0.680 | 86.3% |
| LC80070662014234LGN00 | 0.949 | 99.0% | 0.928 | 98.5% | 0.880 | 97.6% | 0.868 | 97.2% |
| LC80160502014041LGN00 | 0.970 | 98.2% | 0.905 | 94.0% | 0.848 | 89.7% | 0.773 | 83.9% |
| LC80200462014005LGN00 | 0.966 | 98.8% | 0.879 | 95.6% | 0.847 | 93.7% | 0.751 | 89.4% |
| LC80250022014232LGN00 | 0.680 | 86.3% | 0.454 | 55.8% | 0.633 | 82.7% | 0.425 | 51.4% |
| LC80290372013257LGN00 | 0.915 | 95.7% | 0.866 | 92.9% | 0.838 | 91.1% | 0.792 | 88.3% |
| LC80750172013163LGN00 | 0.523 | 99.9% | 0.499 | 98.8% | 0.523 | 99.9% | 0.499 | 98.8% |
| LC80980712014024LGN00 | 0.856 | 90.9% | 0.813 | 87.7% | 0.715 | 79.0% | 0.690 | 76.9% |
| LC81010142014189LGN00 | 0.827 | 91.5% | 0.773 | 88.6% | 0.696 | 84.4% | 0.642 | 81.4% |
| LC81020802014100LGN00 | 0.767 | 89.6% | 0.803 | 91.3% | 0.590 | 81.0% | 0.633 | 82.9% |
| LC81130632014241LGN00 | 0.893 | 97.9% | 0.858 | 97.0% | 0.779 | 94.3% | 0.755 | 93.5% |
| LC81310182013108LGN01 | 0.706 | 98.3% | 0.779 | 98.8% | 0.667 | 97.9% | 0.724 | 98.3% |
| LC81490432014141LGN00 | 0.930 | 100.0% | 0.897 | 100.0% | 0.882 | 99.9% | 0.841 | 99.9% |
| LC81620582014104LGN00 | 0.849 | 98.8% | 0.772 | 97.8% | 0.764 | 97.7% | 0.712 | 96.8% |
| LC81640502013179LGN01 | 0.805 | 95.3% | 0.863 | 97.1% | 0.728 | 92.6% | 0.786 | 95.1% |
| LC81750512013208LGN00 | 0.888 | 93.7% | 0.780 | 85.6% | 0.780 | 86.2% | 0.696 | 77.8% |
| LC81750622013304LGN00 | 0.883 | 95.9% | 0.807 | 93.2% | 0.744 | 90.1% | 0.716 | 89.3% |
| LC81770262013254LGN00 | 0.896 | 98.5% | 0.823 | 97.1% | 0.812 | 96.9% | 0.745 | 95.1% |
| LC81820302014180LGN00 | 0.907 | 99.8% | 0.900 | 99.8% | 0.828 | 99.7% | 0.826 | 99.7% |
| LC81910182013240LGN00 | 0.635 | 99.5% | 0.581 | 99.1% | 0.601 | 99.3% | 0.566 | 98.9% |
| LC81930452013126LGN01 | 0.833 | 92.1% | 0.828 | 90.7% | 0.754 | 87.2% | 0.777 | 86.9% |
| LC82020522013141LGN01 | 0.785 | 94.0% | 0.587 | 75.4% | 0.731 | 92.0% | 0.552 | 71.3% |
| LC82150712013152LGN00 | 0.924 | 95.8% | 0.760 | 84.2% | 0.843 | 90.5% | 0.686 | 77.7% |
| LC82290572014141LGN00 | 0.811 | 88.4% | 0.765 | 85.0% | 0.681 | 78.4% | 0.639 | 75.1% |
| All Scenes | 0.906 | 95.4% | 0.833 | 91.3% | 0.838 | 91.4% | 0.771 | 87.2% |

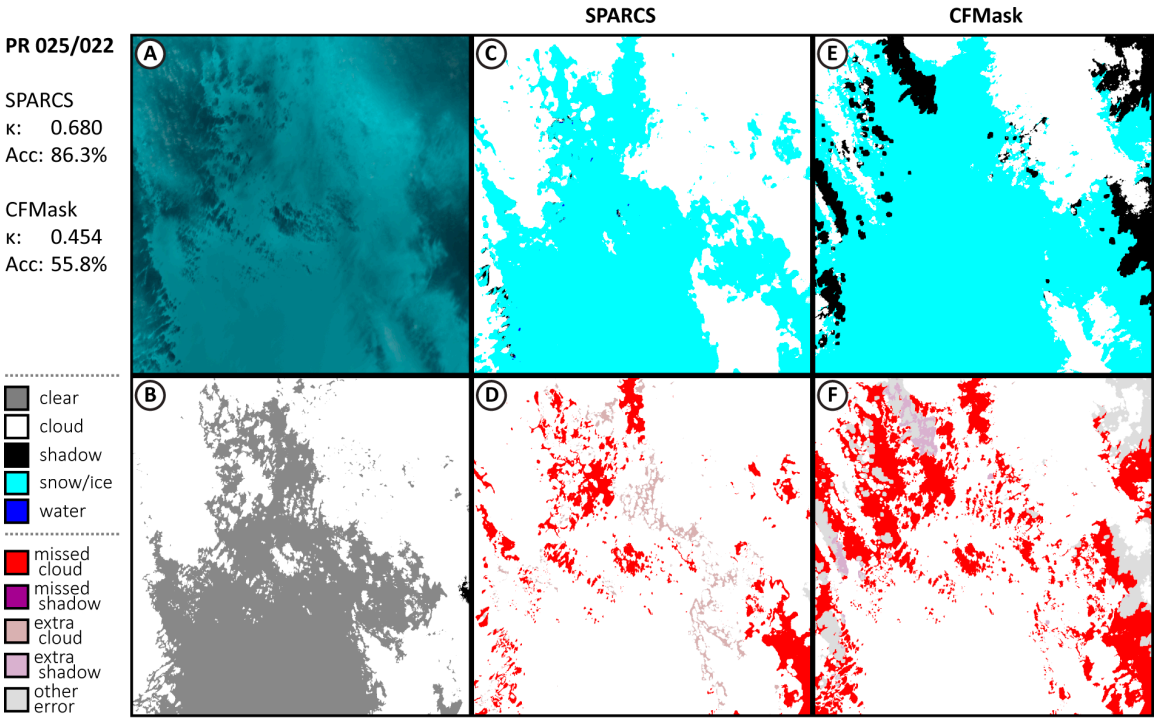


**Figure A7.** Results for Biome test image from PR 098/071, with false color image used during interpretation (**A**), the manually labeled interpretation (**B**), generated masks from SPARCS (**C**) and CFMask (**E**), and respective spatial distribution of errors (**D**,**F**).

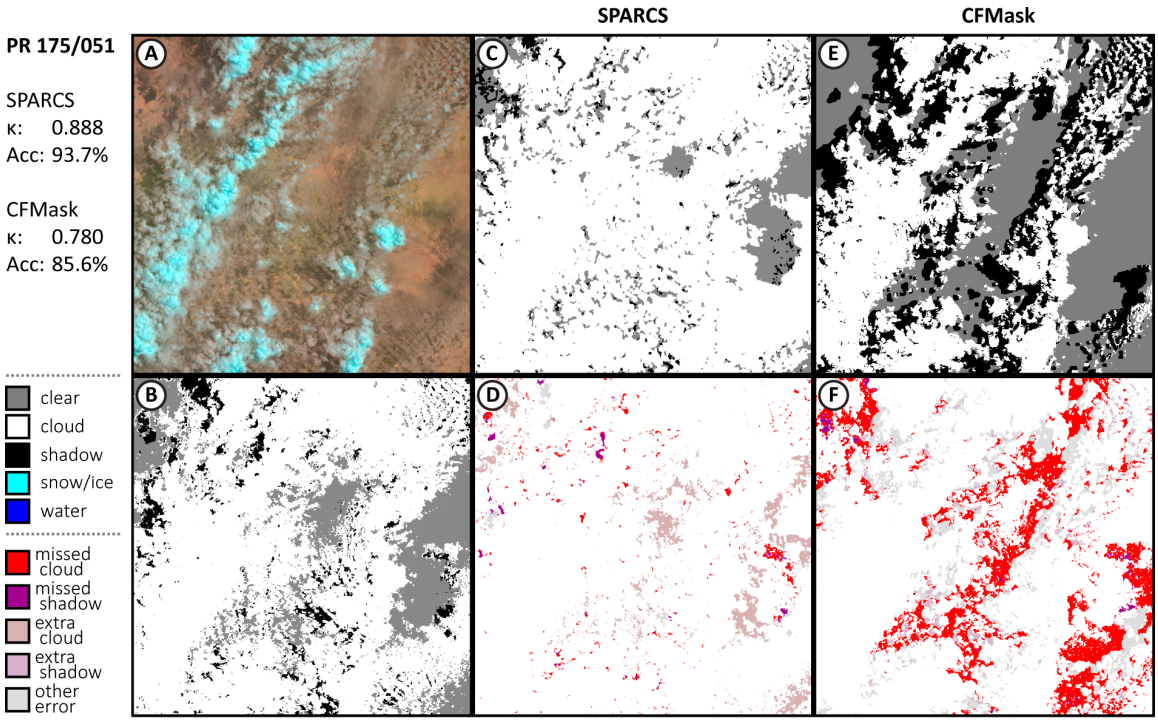


**Figure A8.** Results for Biome test image from PR 175/051, with false color image used during interpretation (**A**), the manually labeled interpretation (**B**), generated masks from SPARCS (**C**) and CFMask (**E**), and respective spatial distribution of errors (**D**,**F**).

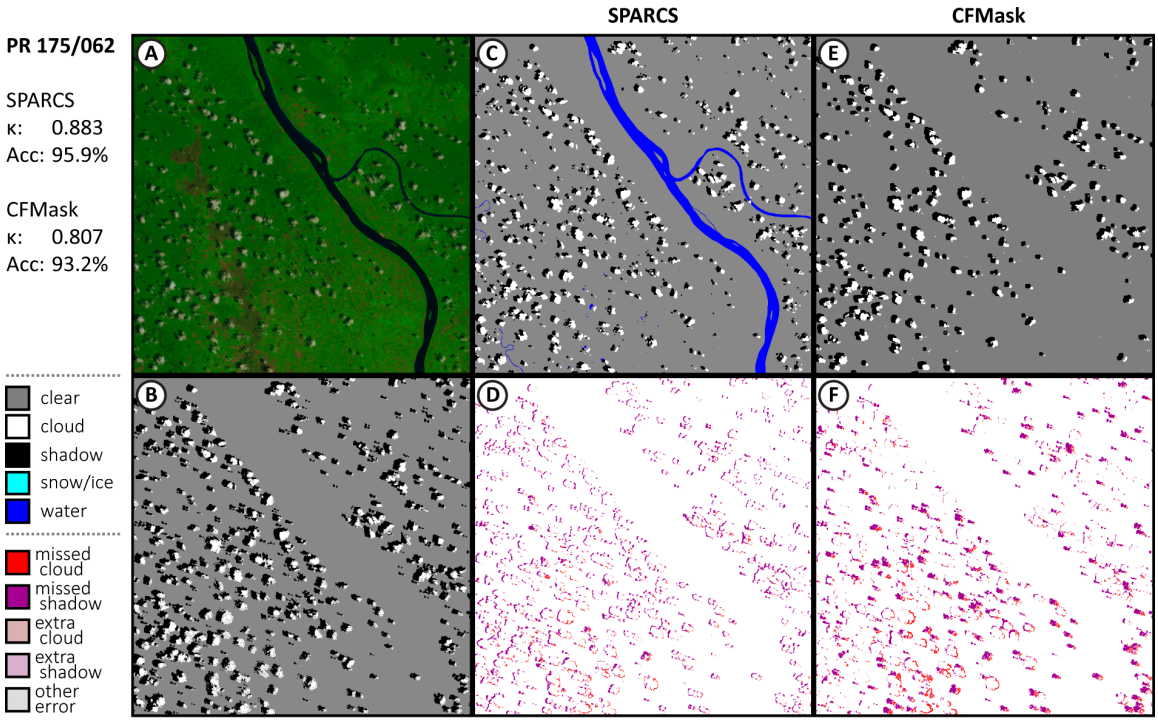


**Figure A9.** Results for Biome test image from PR 175/062, with false color image used during interpretation (**A**), the manually labeled interpretation (**B**), generated masks from SPARCS (**C**) and CFMask (**E**), and respective spatial distribution of errors (**D**,**F**).

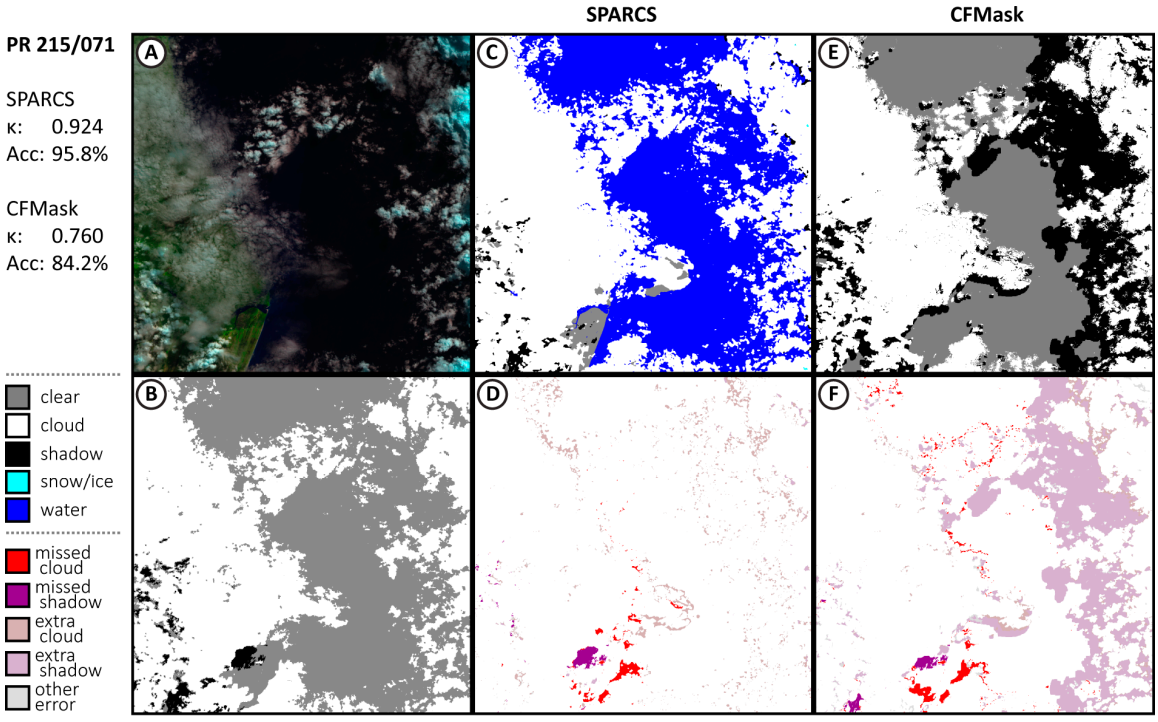


**Figure A10.** Results for Biome test image from PR 215/071, with false color image used during interpretation (**A**), the manually labeled interpretation (**B**), generated masks from SPARCS (**C**) and CFMask (**E**), and respective spatial distribution of errors (**D**,**F**).

## References


1. Wulder, M.A.; Masek, J.G.; Cohen, W.B.; Loveland, T.R.; Woodcock, C.E. Opening the archive: How free data has enabled the science and monitoring promise of Landsat. *Remote Sens. Environ.* **2012**, *122*, 2–10. [CrossRef]
2. Wulder, M.A.; Loveland, T.R.; Roy, D.P.; Crawford, C.J.; Masek, J.G.; Woodcock, C.E.; Allen, R.G.; Anderson, M.C.; Belward, A.S.; Cohen, W.B.; et al. Current status of Landsat program, science, and applications. *Remote Sens. Environ.* **2019**, *225*, 127–147. [CrossRef]
3. Ju, J.; Roy, D.P. The availability of cloud-free Landsat ETM+ data over the conterminous United States and globally. *Remote Sens. Environ.* **2008**, *112*, 1196–1211. [CrossRef]
4. Kennedy, R.E.; Yang, Z.; Braaten, J.; Copass, C.; Antonova, N.; Jordan, C.; Nelson, P. Attribution of disturbance change agent from Landsat time-series in support of habitat monitoring in the Puget Sound region, USA. *Remote Sens. Environ.* **2015**, *166*, 271–285. [CrossRef]
5. Cohen, W.; Healey, S.; Yang, Z.; Stehman, S.; Brewer, C.; Brooks, E.; Gorelick, N.; Huang, C.; Hughes, M.; Kennedy, R. How similar are forest disturbance maps derived from different Landsat time series algorithms? *Forests* **2017**, *8*, 98. [CrossRef]
6. Healey, S.P.; Cohen, W.B.; Yang, Z.; Brewer, C.K.; Brooks, E.B.; Gorelick, N.; Hernandez, A.J.; Huang, C.; Hughes, M.J.; Kennedy, R.E. Mapping forest change using stacked generalization: An ensemble approach. *Remote Sens. Environ.* **2018**, *204*, 717–728. [CrossRef]
7. Hollingsworth, B.V.; Chen, L.; Reichenbach, S.E.; Irish, R.R. Automated cloud cover assessment for Landsat TM images. *Proc. SPIE* **1996**, *2819*, 170–179.
8. Zhu, Z.; Woodcock, C.E. Object-based cloud and cloud shadow detection in Landsat imagery. *Remote Sens. Environ.* **2012**, *118*, 83–94. [CrossRef]
9. Hughes, M.; Hayes, D. Automated detection of cloud and cloud shadow in single-date Landsat imagery using neural networks and spatial post-processing. *Remote Sens.* **2014**, *6*, 4907–4926. [CrossRef]
10. Foga, S.; Scaramuzza, P.L.; Guo, S.; Zhu, Z.; Dilley, R.D., Jr.; Beckmann, T.; Schmidt, G.L.; Dwyer, J.L.; Hughes, M.J.; Laue, B. Cloud detection algorithm comparison and validation for operational Landsat data products. *Remote Sens. Environ.* **2017**, *194*, 379–390. [CrossRef]
11. Rawat, W.; Wang, Z. Deep convolutional neural networks for image classification: A comprehensive review. *Neural Comput.* **2017**, *29*, 2352–2449. [CrossRef] [PubMed]
12. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
13. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. In Proceedings of the Advances in Neural Information Processing Systems, Lake Tahoe, NV, USA, 3–6 December 2012; pp. 1097–1105.
14. Shin, H.-C.; Roth, H.R.; Gao, M.; Lu, L.; Xu, Z.; Nogues, I.; Yao, J.; Mollura, D.; Summers, R.M. Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning. *IEEE Trans. Med. Imaging* **2016**, *35*, 1285–1298. [CrossRef] [PubMed]
15. Wei, Y.; Xia, W.; Huang, J.; Ni, B.; Dong, J.; Zhao, Y.; Yan, S. Cnn: Single-label to multi-label. *arXiv* **2014**, arXiv:1406.5726.
16. Shi, M.; Xie, F.; Zi, Y.; Yin, J. Cloud detection of remote sensing images by deep learning. In Proceedings of the 2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), Beijing, China, 10–15 July 2016; pp. 701–704.
17. Xie, F.; Shi, M.; Shi, Z.; Yin, J.; Zhao, D. Multilevel cloud detection in remote sensing images based on deep learning. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2017**, *10*, 3631–3640. [CrossRef]
18. Kussul, N.; Lavreniuk, M.; Skakun, S.; Shelestov, A. Deep learning classification of land cover and crop types using remote sensing data. *IEEE Geosci. Remote Sens. Lett.* **2017**, *14*, 778–782. [CrossRef]
19. Mateo-García, G.; Gómez-Chova, L.; Camps-Valls, G. Convolutional neural networks for multispectral image cloud masking. In Proceedings of the 2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), Fort Worth, TX, USA, 23–28 July 2017; pp. 2255–2258.
20. Zhan, Y.; Wang, J.; Shi, J.; Cheng, G.; Yao, L.; Sun, W. Distinguishing cloud and snow in satellite images via deep convolutional network. *IEEE Geosci. Remote Sens. Lett.* **2017**, *14*, 1785–1789. [CrossRef]

21. Drönner, J.; Korfhage, N.; Egli, S.; Mühling, M.; Thies, B.; Bendix, J.; Freisleben, B.; Seeger, B. Fast Cloud Segmentation Using Convolutional Neural Networks. *Remote Sens.* **2018**, *10*, 1782. [CrossRef]
22. Liu, C.-C.; Zhang, Y.-C.; Chen, P.-Y.; Lai, C.-C.; Chen, Y.-H.; Cheng, J.-H.; Ko, M.-H. Clouds Classification from Sentinel-2 Imagery with Deep Residual Learning and Semantic Image Segmentation. *Remote Sens.* **2019**, *11*, 119. [CrossRef]
23. Wieland, M.; Li, Y.; Martinis, S. Multi-sensor cloud and cloud shadow segmentation with a convolutional neural network. *Remote Sens. Environ.* **2019**, *230*, 111203. [CrossRef]
24. Li, Z.; Shen, H.; Cheng, Q.; Liu, Y.; You, S.; He, Z. Deep learning based cloud detection for medium and high resolution remote sensing images of different sensors. *ISPRS J. Photogramm. Remote Sens.* **2019**, *150*, 197–212. [CrossRef]
25. Shao, Z.; Pan, Y.; Diao, C.; Cai, J. Cloud Detection in Remote Sensing Images Based on Multiscale Features-Convolutional Neural Network. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 4062–4076. [CrossRef]
26. Yang, J.; Guo, J.; Yue, H.; Liu, Z.; Hu, H.; Li, K. CDnet: CNN-Based Cloud Detection for Remote Sensing Imagery. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 6195–6211. [CrossRef]
27. Chai, D.; Newsam, S.; Zhang, H.K.; Qiu, Y.; Huang, J. Cloud and cloud shadow detection in Landsat imagery based on deep convolutional neural networks. *Remote Sens. Environ.* **2019**, *225*, 307–316. [CrossRef]
28. Jeppesen, J.H.; Jacobsen, R.H.; Inceoglu, F.; Toftegaard, T.S. A cloud detection algorithm for satellite imagery based on deep learning. *Remote Sens. Environ.* **2019**, *229*, 247–259. [CrossRef]
29. Zeiler, M.D.; Krishnan, D.; Taylor, G.W.; Fergus, R. Deconvolutional networks. In Proceedings of the 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2010), San Francisco, CA, USA, 13–18 June 2010; p. 7.
30. Long, J.; Shelhamer, E.; Darrell, T. Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 3431–3440.
31. Jouppi, N.P.; Young, C.; Patil, N.; Patterson, D.; Agrawal, G.; Bajwa, R.; Bates, S.; Bhatia, S.; Boden, N.; Borchers, A. In-datacenter performance analysis of a tensor processing unit. In Proceedings of the 2017 ACM/IEEE 44th Annual International Symposium on Computer Architecture (ISCA), Toronto, ON, Canada, 24–28 June 2017; pp. 1–12.
32. Olson, D.M.; Dinerstein, E. The Global 200: Priority ecoregions for global conservation. *Ann. Mo. Bot. Gard.* **2002**, *89*, 199–224. [CrossRef]
33. Tompson, J.; Goroshin, R.; Jain, A.; LeCun, Y.; Bregler, C. Efficient object localization using convolutional networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 648–656.
34. USGS; EROS; NASA. Landsat 8 (L8) Data User's Handbook Version 4. 2019. Available online: https://prd-wret.s3-us-west-2.amazonaws.com/assets/palladium/production/atoms/files/LSDS-1574_L8_Data_Users_Handbook_v4.pdf (accessed on 25 April 2019).
35. Data Input Pipeline Performance. Available online: https://www.tensorflow.org/guide/performance/datasets (accessed on 21 August 2019).
36. Kullback, S.; Leibler, R.A. On information and sufficiency. *Ann. Math. Stat.* **1951**, *22*, 79–86. [CrossRef]
37. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
38. Cohen, J. A coefficient of agreement for nominal scales. *Educ. Psychol. Meas.* **1960**, *20*, 37–46. [CrossRef]
39. Yu, F.; Koltun, V. Multi-scale context aggregation by dilated convolutions. *arXiv* **2015**, arXiv:1511.07122.