Contents lists available at ScienceDirect

Remote Sensing of Environment

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

A cloud detection algorithm for satellite imagery based on deep learning

Jacob Høxbroe Jeppesen[a,*], Rune Hylsberg Jacobsen[a], Fadil Inceoglu[a,b,c], Thomas Skjødeberg Toftegaard[a]

[a] Department of Engineering, Aarhus University, Finlandsgade 22, 8200 Aarhus N., Denmark
[b] Department of Geoscience, Aarhus University, Høegh-Guldbergs Gade 2, 8000 Aarhus C., Denmark
[c] Stellar Astrophysics Centre, Department of Physics and Astronomy, Aarhus University, Ny Munkegade 120, 8000 Aarhus C., Denmark

ARTICLE INFO

Keywords:
Cloud detection
Optical satellite imagery
Deep learning
Open data

ABSTRACT

Reliable detection of clouds is a critical pre-processing step in optical satellite based remote sensing. Currently, most methods are based on classifying invidual pixels from their spectral signatures, therefore they do not incorporate the spatial patterns. This often leads to misclassifications of highly reflective surfaces, such as human made structures or snow/ice. Multi-temporal methods can be used to alleviate this problem, but these methods introduce new problems, such as the need of a cloud-free image of the scene. In this paper, we introduce the Remote Sensing Network (RS-Net), a deep learning model for detection of clouds in optical satellite imagery, based on the U-net architecture. The model is trained and evaluated using the Landsat 8 Biome and SPARCS datasets, and it shows state-of-the-art performance, especially over biomes with hardly distinguishable scenery, such as clouds over snowy and icy regions. In particular, the performance of the model that uses only the RGB bands is significantly improved, showing promising results for cloud detection with smaller satellites with limited multi-spectral capabilities. Furthermore, we show how training the RS-Net models on data from an existing cloud masking method, which are treated as noisy data, leads to increased performance compared to the original method. This is validated by using the Fmask algorithm to annotate the Landsat 8 datasets, and then use these annotations as training data for regularized RS-Net models, which then show improved performance compared to the Fmask algorithm. Finally, the classification time of a full Landsat 8 product is 18.0 ± 2.4 s for the largest RS-Net model, thereby making it suitable for production environments.

1. Introduction

Remote sensing has gained immense attention in the last decade, in particular following the open data policy of the Landsat satellites from NASA in 2008. ESA further increased the open data availability of satellite data with the Copernicus programme, which started to provide free satellite data, including optical imagery and radar measurements of the Earth's surface, and chemical composition measurements of the troposphere. It is expected that these data will be combined with modern data analytics tools to stimulate innovative and economic growth. One vision of the Copernicus programme is to extend the use of remote sensing by the general population, much like the GPS is nowadays used excessively on regular smartphones. In addition to the publicly available high-quality data, other facilitators have significantly decreased the cost of employing satellite data, such as the increase in Internet bandwidth, storage capacity, processing power, and the development of sophisticated open source software tools. Although large improvements in software for data pre-processing and analytics have been carried out recently, there are still issues to be solved in a range of areas. In particular, automated classification is crucial in many use cases, such as harvest yield estimation (Prasad et al., 2006; Ferencz et al., 2004), change detection (Verbesselt et al., 2010; Sakamoto et al., 2005), and disaster management (Voigt et al., 2007; Tralli et al., 2005; Joyce et al., 2009). Cloud coverage often disturb analyses, and the annual mean global cloud cover is estimated as approximately 66% (Zhang et al., 2004). Therefore, it is a vital pre-processing step to correctly and efficiently classify clouds, before the satellite imagery can be used for further analysis. Current methods primarily rely on single-pixel based classification algorithms, thus mainly focusing on the spectral signature. This leads to misclassifications of pixels with similar spectral signatures, for example, highly reflective human made structures, sand in deserts, and snow/ice. The spatial patterns are often ignored, or solely used in a simple post-processing step, mainly due to the lack of efficient methods for including them in the analysis. Deep learning algorithms have gained momentum in recent years, and provide unprecedented performance for combining spatial and spectral patterns

* Corresponding author.
E-mail address: jhj@eng.au.dk (J.H. Jeppesen).

https://doi.org/10.1016/j.rse.2019.03.039
Received 29 August 2018; Received in revised form 28 March 2019; Accepted 30 March 2019
Available online 24 May 2019

for classification tasks, which make them highly suitable for remote sensing classification tasks. The underlying reason for the improvement in performance is the ability to include the feature extraction in the optimization loop, thus improving this step immensely compared to earlier hand-crafted feature extraction methods. The result is a larger machine learning model, which applies a series of non-linear transformations to the input data, such that an optimal data representation is found as input of the actual classifier, constituting the latter part of the model. Deep learning models introduce new obstacles however, such as the need for large training datasets to achieve high performing models.

In this paper, we introduce Remote Sensing Network (RS-Net), a deep learning model based on the U-net architecture for cloud classification, that shows state-of-the-art performance on the Landsat 8 Biome and SPARCS cloud cover validation datasets. The contributions include a cloud detection algorithm which incorporates the spatial patterns, thereby achieving high performance, even when the classification is based on a single spectral band. We provide a simplistic approach, leading to a few requirements for pre-processing, fast convergence, fast inference due to the use of a fully convolutional neural network approach, and state-of-the-art performance. An evaluation on the Landsat 8 datasets is presented, and the performance of RS-Net is compared to the Fmask algorithm. Subsequently, it is shown how a regularized RS-Net model can be trained on annotations produced by the Fmask algorithm, and then show improved performance over said algorithm. Finally, a brief timing analysis shows how the processing time for the classification is 18.0 ± 2.4 s per Landsat 8 product on a regular prosumer PC. The hardware and software setups used for all experiments is given in Section 3, and elaborated in Appendix A. The code is publicly available (https://github.com/JacobJeppesen/RS-Net). We plan to extend RS-Net beyond cloud detection, including methods for semi-supervised and multi-temporal classification, as well as methods for improved interpretability of the classification.

2. Background

The NASA Landsat 8 satellite is an 11 band multi-spectral satellite with down to 15 m resolution and a revisit time of 16 days. It comprises two instruments, the Operational Land Imager (OLI) and the Thermal Infrared Sensor (TIRS). The OLI provides 9 of the 11 bands, and covers the wavelengths from 0.435 μm to 2.294 μm, with one wide panchromatic band, providing 15 m resolution, compared to the 30 m resolution of the remaining bands (https://landsat.usgs.gov/what-are-band-designations-landsat-satellites, n.d.). Band 9 is specifically designed to detect cirrus clouds, which often contaminate satellite imagery with their semi-transparent nature. The TIRS instrument covers the wavelengths from 10.60 μm to 12.51 μm in the remaining two bands, providing surface temperatures. The high spatial resolution combined with the multi-spectral capabilities has proved to be highly valuable in monitoring our environment, and the open data policy from NASA has significantly increased the use of satellite data in academia and the industry. In addition to the NASA satellites, ESA has decided to complement the open data policy by launching a satellite programme with the main purpose of providing high quality satellite data to governments, academia, and industry. The ESA Sentinel-2 satellites are two identical 13 band multi-spectral satellites with down to 10 m resolution, operating in opposite sides of the orbit to provide a revisiting time of 5 days. These satellites provide the basis for one of the largest publicly available datasets produced, already in excess of 25 PB (Zhu et al., 2017).

One of the first pre-processing steps required when employing optical satellite data is to perform cloud masking, where each cloud pixel in the scene is detected, as it is critical for further analysis. With the recent advances in data availability and processing power, automated analysis of satellite data with no human in the loop is increasingly used, making misclassifications even more critical. The most widely used methods for cloud detection are compared in (Foga et al., 2017), where several methods are evaluated on manually annotated Landsat 7 and 8 datasets. The optimal method is found to be the Fmask algorithm (Zhu and Woodcock, 2012; Zhu et al., 2015), which they then implement in the C programming language to obtain a 90% reduction in runtime. It classifies the clouds based on the spectral characteristics, and then detects the cloud shadows based on geometry using an estimated physical size of the cloud (including height) and the sun angle. It uses a decision tree with a range of rules to first establish two potential cloud layers, which are then combined to produce a final cloud mask. It tries to overcome single pixel misclassifications by filtering such that a pixel can only be classified as cloud if at least five pixels in the 3-by-3 pixels neighborhood are classified as clouds. A buffer around all clouds are then added, as it is preferred to lose data rather than accidentally including clouds in the analysis. Importantly, it is noted in (Foga et al., 2017) how (Scaramuzza et al., 2012) found a 7% error in the manual annotation between human analysts. The underlying issue is that clouds are often semi-transparent, and the definition of the level of transparency which constitutes a cloud varies from person to person. Their solution is to only use one single analyst when they create the Landsat 8 Biome dataset, thus it is expected that discrepancies of the definition of a cloud varies in the ground truth annotations between the datasets. Existing machine learning based methods, such as the neural networks approach investigated in (Hughes and Hayes, 2014), were found to perform sub par compared to the Fmask algorithm (Zhu, 2017). Deep learning, or deep artificial neural networks, is a branch of machine learning where the feature extraction has been automated, which has gained momentum in recent years. Convolutional Neural Networks (CNNs) have proven to be effective within computer vision, such as the present task on cloud detection. The main improvement over previous methods is the automated feature extraction, which was previously a manual task. In practice, the CNN builds a hierarchy of data representations, such that input to the classifier is its optimal representation. Semantic segmentation is the task of assigning each pixel of an input image to a specific class, which is an area where deep learning has shown particularly good results. These methods suit the remote sensing domain extraordinarily well, and often show high accuracy with low processing time. Additionally, they perform well under different lighting conditions, a major source of noise in remote sensing. Although deep learning methods have not yet been widely adopted by the remote sensing field, there is an exponential increase in the number of studies regarding the subject (Zhu et al., 2017). Deep learning has been used for various remote sensing tasks, such as road detection (Zhang et al., 2018), sea-land detection (Li et al., 2018a), and land cover mapping (Karakizi et al., 2018). A deep learning method based on PCANet was used on the Landsat 8 Biome dataset to perform cloud cover detection (Zi et al., 2018). They split the dataset into 24 images for training and 72 images for testing, obtaining an accuracy, precision, recall, and F_1 score of 91.16, 89.14, 89.33, and 89.23, respectively. They implemented the method in MATLAB, reporting an average cloud detection time per Landsat 8 product of 12.8 min for their final proposed framework. In (Mateo-García et al., 2018), a multi-temporal approach to cloud masking was tested on the same Landsat 8 datasets employed in this paper, showing improvements compared to the FMask algorithm. In essence, the cloud detection task is redefined as a change detection task, with a clear background image as part of the analysis. They report an accuracy of 94.13 on the Landsat 8 Biome dataset. A multi-level cloud detection algorithm based on CNNs was proposed by (Chen et al., 2018). The input image is first divided into superpixels comprised of clusters of pixels with similar spectral signatures. These are then fed to multiple CNNs to classify the superpixels. A method based on the U-net architecture was used by (Drönner et al., 2018) to perform cloud detection on Meteosat Second Generation satellite imagery, implemented in Caffe, reporting both high performance and low inference time. In (Le Goff et al., 2017), CNNs are used for cloud detection on SPOT6 imagery, showing improved performance over traditional methods. One technical challenge regarding deep learning, however, is the training of classifiers with limited annotated data. One direction for solving this challenge is through semi-supervised learning, which aims for improving the supervised learning task by the additional use of unlabeled data. Advanced methods have been shown to significantly improve the performance of a multi-spectral classifier by co-training with hyperspectral data (Hong et al., 2019). This is particularly useful in remote sensing, due to the vast amount of available data.

3. The RS-Net architecture

This section will provide a brief introduction to the deep learning algorithm applied, followed by an overview of the network architecture, the datasets, implementation, and training procedure. A comprehensive introduction to deep learning is beyond the scope of this paper, however, the following section is intended to provide a sufficiently detailed overview of the algorithm to obtain an understanding of the underlying mechanisms.

3.1. The deep learning model

We want to map an input image, $\mathbf{X} \in \mathbb{R}^{w \times h \times c}$, to a pixel-wise classification map, $\mathbf{Y} \in \mathbb{R}^{w \times h \times 1}$, where w is the width, h is the height, and c is the number of input channels. The output classification provides a pixel-wise confidence metric from 0 to 1 for a cloud being present. A fully convolutional network can be described as a mapping function, f, given by:

$$\mathbf{Y} = f(\mathbf{X}; \theta), \tag{1}$$

where $\boldsymbol{\theta}$ are parameters of the mapping function. By forming the function f from the composition of two functions, f_1 and f_2, we can group the parameters, such that

$$\mathbf{Y} = f_2(f_1(\mathbf{X}; \theta_1); \theta_2), \tag{2}$$

thus the mapping function becomes a composition of smaller sub-functions, or layers. As seen in Fig. 1, the final model is a composition of many layers, thus constituting a *deep* model. Most important are the convolutional layers, which produce feature maps by convolving their input with a specified number of kernels:

$$(\mathbf{I} * \mathbf{K}_d)(i, j, d) = \sum_m \sum_n \sum_l \mathbf{I}(i - m, j - n, l)\mathbf{K}_d(m, n, l) \tag{3}$$

where $\mathbf{I}$ is the input and $\mathbf{K}_d$ the d'th m by n by l kernel. The number of kernels, d, depends on the depth of the output feature maps, and it is chosen manually when designing the network architecture. It should be noted that the kernel is used on a small region spatially, but always works on the entire depth of the input feature map. The kernels perform a linear transformation, followed by a non-linearity, referred to as an activation function. One example of such is the often used rectified linear unit (ReLU) function, which zeroizes all negative values, such that

$$\mathbf{K}_d(\mathbf{X}) = max(0, \mathbf{W}_d\mathbf{X} + b_d), \tag{4}$$

where the weights, $\mathbf{W}_d$, and bias, b_d, are trainable parameters in $\boldsymbol{\theta}$.

In the final convolutional layer, however, the sigmoid function is used as activation function, as it is used to output the predicted cloud probabilities. To train the model, the binary cross-entropy function is used to calculate a loss based on the true cloud mask and the predicted cloud mask:

$$\mathcal{L}(\widehat{\mathbf{Y}}, \mathbf{Y}) = -\frac{1}{w \cdot h} \sum_w \sum_h \Big(\widehat{\mathbf{Y}}_{w,h} \ln(\mathbf{Y}_{w,h}) + (1 - \widehat{\mathbf{Y}}_{w,h}) \ln(1 - \mathbf{Y}_{w,h})\Big), \tag{5}$$

where $\hat{\mathbf{Y}}$ is the true cloud mask and $\mathbf{Y}$ is the predicted cloud mask. This loss is then used to calculate the gradient on each parameter in $\boldsymbol{\theta}$ through the back-propagation algorithm (Rumelhart et al., 1986) before a gradient descent algorithm is employed for the optimization task. The advantage of a convolutional neural network is that the kernel parameters are re-used over the entire input, rather than having individual connections to each neuron in the preceding layer. This results in significantly fewer parameters in the final model than a fully connected layer, as known from traditional multi-layer perceptrons. The consecutive convolutional layers hierarchically build an improved data representation, as the trainable kernels effectively learn the optimal feature extraction strategy. This allows the latter part of the network, which is responsible for the actual classification, to perform optimally. Although the loss function is non-convex, empirical results show that the gradient descent algorithm converges at high-performing local minima, and that the global minimum often leads to overfitting (Choromanska et al., 2015). It should be noted that changing the model to a multi-class predictor is straight-forward, and simply requires the activation function of the final convolutional layer to be changed to a softmax function and the loss function to be changed to categorical cross-entropy.

In addition to the convolutional layers, the max-pooling layers are used for downsampling the feature maps. This is simply executed by dividing the feature maps spatially into 2-by-2 blocks and passing the maximum value, thereby discarding 75% of the feature map. In Fig. 1, it

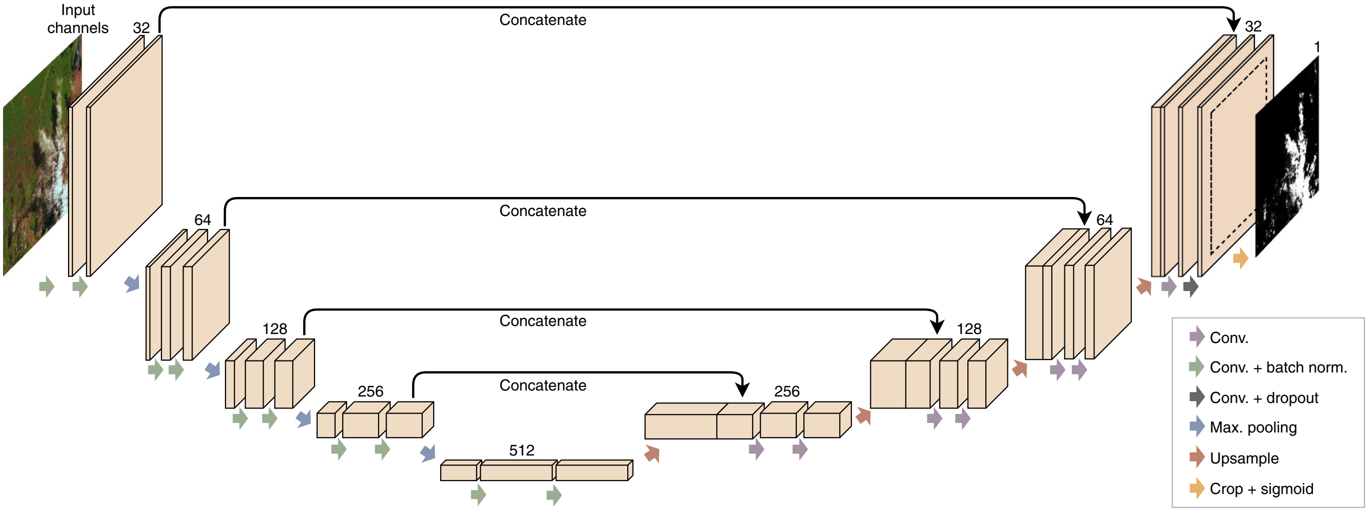


Fig. 1. The RS-Net architecture (based on U-net (Ronneberger et al., 2015)), where the depth of the feature maps in the stages are provided by the number above each stage.

can be seen how the spatial size is decreased in the first half of the model, referred to as the encoder, as a consequence of the max-pooling operation. In the latter half of the model, referred to as the decoder, the feature maps are upsampled, simply by copying each value into a 2-by-2 grid spatially. The resulting design of the model is referred to as the network architecture. There are numerous solutions for pixel-wise classification, such as FCN (Long et al., 2015), SegNet (Badrinarayanan et al., 2017), and U-net (Ronneberger et al., 2015). The RS-Net is based on the U-net architecture, which was originally designed for biomedical applications (Ronneberger et al., 2015). It should be noted that the depth of the convolutional layers are normally doubled after each max-pooling layer, and halved after each upsampling layer. The encoder uses stages of convolutional layers and a max pooling layer, until the fourth stage, after which the upsampling process by the decoder starts. Each stage in the decoder begins with an upsampling of the feature map from the previous stage, concatenated by the feature maps from the equivalent stage in the encoder, which compensates for the lost information in the max pooling layer. The model is regularized to avoid over-fitting by using L2-regularization, dropout, and batch normalization layers. L2-regularization adds the squared sums of the weights to the loss function, thereby preventing individual neurons to dominate (Ng, 2004), and is used in all convolutional layers. Dropout is a method where neurons randomly have their outputs set to zero (Srivastava et al., 2014). This effectively forces the data to take different paths through the network, thus preventing overfitting. Batch normalization is a method for feature normalization throughout the network (Ioffe and Szegedy, 2015), which is applied after each convolutional layer in the encoder. Compared to the original U-net architecture, the introduction of batch normalization significantly reduces the training time and the requirements to the pre-processing step, which will be briefly discussed in the following section. Combining batch normalization and dropout introduce issues when used together, thus following the advices in (Li et al., 2018b), we only use dropout after the last batch normalization layer in the entire architecture, instead of after each convolutional block. A cropping layer was inserted after the final layer to discard the outermost regions of the patch, which perform poorly due to the lack of spatial information in regions near the borders of the image. In addition, the clipping layer reduces training time, as the error-prone gradients from the border of the image is not fed back into the network. Finally, the depth of the feature maps are halved, thereby reducing both training and inference time.

The model can be trained on a desired number of spectral bands, thus it is straightforward to investigate the performance of the model for several different band combinations. This is relevant as many satellites, particularly low-cost nano-satellites, do not have the multispectral capabilities of, for example, the Landsat 8 satellite. Therefore, RS-Net will be investigated based on its performance on five different band combinations, with (i) all available bands, (ii) all available bands except thermal (i.e., exclude band 10 and 11), (iii) the red/green/blue/infrared (RGBI) bands, (iv) the red/green/blue (RGB) bands, and (v) the green (G) band alone. Band 8 (the panchromatic band) is not used in this work, and the remaining bands have not been pansharpened.

3.2. Implementation and training procedure

The implementation of RS-net was carried out in Keras (2.1.6) with TensorFlow (1.9.0) as backend, with the software versions as described Appendix A. This setup has been used for all experiments presented in this paper. The Fmask algorithm automatically adds a buffer around classified clouds, which have been disabled here to obtain comparable results. Adding a buffer around the cloud pixels is a simple method to avoid false negatives where the transition from cloud to clear is difficult to determine. It is a post-processing step which will lead to an equal improvement for both methods, thus it will not be further investigated in this paper.

All input data is normalized to values between 0 and 1, simply done by dividing their values by 65,535, as this is the maximum value of the 16 bit integer constituting the original format of the data. Other normalization strategies were investigated, such as normalizing the individual bands based on the statistics from the Biome dataset, and normalization to values between -1 and 1, which is more sensible taken the initialization of the parameters in the model into consideration. But the batch normalization layers made this superfluous, thus the simplest approach was used. Although it would be beneficial to classify the entire image concurrently to make use of the spatial patterns of the entire scene, the hardware introduce limitations. Therefore, the image must be divided into patches which are classified separately, and then stitched together to form the final classification. The size of these patches has a significant influence on the final accuracy, as it is on the one hand beneficial to use as large patches as possible to catch spatial patterns, but on the other hand, an increase in the patch size decreases the batch size during training. This heavily influences the batch normalization layers, thus resulting in a balance between a sufficiently large patch size for incorporating spatial patterns and a large batch size for improving the regularization during training. As previously mentioned, a clipping layer was introduced to avoid border issues, thereby creating an overlap of 40 pixels on the predicted patches. This results in issues at the borders of the entire image, where there is a sudden transition to pixels with a value of zero. If the satellite image constitutes a square, such as imagery from the SPARCS dataset, this is easily dealt with by mirroring the data at the borders, such that the scene is padded with a reflection of the outermost pixels. This is rarely the case for satellite imagery, as the borders are often tilted with black around it, which is also seen in the Biome dataset (see Fig. 8). This is due to the imagery being projected onto a standardized projection, and thus, simple mirroring cannot be carried out. To solve this issue, the black pixels are band-wise inpainted by the mean value of the non-zero pixels in the remainder of the specific patch being processed. A more advanced inpainting method (Telea, 2004) was tested, but did not result in significant benefit.

Data augmentation is the concept of altering the training data to effectively increase the amount of training examples. In this work, the training patches were given a 50% chance of a horizontal flip followed by a 50% chance of a vertical flip. This effectively increases the training data size by a factor of 4, as the model perceive these flipped images as new images. This was particularly relevant when using the SPARCS dataset for training, due to the relatively small size of the dataset. Additionally, the training data patches were overlapping by 120 pixels. The AMSGrad variant (Reddi et al., 2018) of the Adam optimizer (Kingma et al., 2015) was employed for training, and hyper-parameter optimization was first done semi-manually using a simple grid-search algorithm. The advantage of this is the improved transparency in how the individual hyperparameters affect the performance. Through grid search, the activation functions were decided to be exponential linear units (Clevert et al., 2015), batch normalization was decided to be used on each convolutional layer in the encoder with a momentum of 0.7, and the initialization was decided to be Xavier normal initialization (Glorot and Bengio, 2010). Subsequently, random search was used for finding the optimal model, by tuning the learning rate, dropout probability, L2-regularization, and the number of training epochs. This random sampling of hyperparameters, with subsequent local optimization using the Adam algorithm, results in specific configurations of RS-Net exhibiting the highest performance.

4. Landsat 8 Biome and SPARCS datasets

The 195 GB Landsat 8 Biome dataset consists of 96 Landsat 8 scenes, which were gathered and annotated by (Foga et al., 2017), with an emphasis on being globally representative, and designed such that it could easily be split in two for validation and training data in machine learning models. It is divided into 8 different biomes, that are barren, forest, grass/crops, shrubland, urban, water, and wetlands, and further divided into 4 classes, namely ‘cloud’, ‘thin cloud’, ‘cloud shadow’, and ‘clear’. It should be noted, however, that shadows are only annotated in 30 of the 96 scenes. Additionally, the scenes are divided into groups of images with little or no clouds ($< 35\%$), mid-cloudy ($\geq 35\%$ and $\leq 65\%$), and cloudy ($> 65\%$). The smaller 1.6 GB Landsat 8 SPARCS dataset (Hughes and Hayes, 2014) consists of 80 1000 × 1000 pixels scenes. It is divided into 7 classes, namely ‘cloud shadow’, ‘cloud shadow over water’, ‘water’, ‘snow’, ‘land’, ‘cloud’, and ‘flooded’. To investigate the agreement between analysts on the annotation of the scenes of the SPARCS dataset, an additional analyst made an annotations of 6 sub-scenes, showing 96% agreement, thus again showing the ambiguity in annotating these dataset (Foga et al., 2017).

Both datasets were Top of Atmosphere (ToA) corrected, based on the ToA algorithm in the Python Fmask implementation (http://pythonfmask.org/en/latest/, n.d.). A single product from the SPARCS dataset (LC80010812013365LGN00_18) lead to issues when calculating the pixel-wise sun angle in the ToA algorithm, and was therefore processed manually using the centroid sun angle. Then the class distributions in the datasets were compared, as they should ideally be similar and representative of the general cloud coverage in Landsat 8 scenery. To achieve this, some classes in the two datasets were combined, for example, ‘cloud shadow’ and ‘cloud shadow over water’ were combined to ‘cloud shadow’ in the SPARCS dataset, and ‘thin cloud’ and ‘cloud’ were combined to ‘cloud’ in the Biome dataset. This resulted in the comparative classes, ‘clear’, ‘cloud’, and ‘shadow’, where possible discrepancies in the distributions between the two datasets could be evaluated.

The mean coverage for each of the three classes were calculated in each scene. The distributions are not similar, and the SPARCS dataset contains more clear scenes (Fig. 2). The distributions also show that the Biome dataset has been collected with a strong emphasis on the three groups of clear, mid-cloudy, and cloudy scenes, which is probably not representative for Landsat 8 data in general (Fig. 2). The smaller amount of cloud shadows in the Biome dataset compared to the SPARCS dataset is due to the fact that not all scenes were annotated with cloud shadows, and that the Biome dataset contains many more scenes completely covered with clouds, where no shadows are present. The lack of cloudy images in the SPARCS dataset makes it difficult to evaluate the performance on scenes completely covered by clouds. Therefore, using the Biome dataset, which does not have the same drawbacks and additionally is much larger, results in more reliable evaluations of the performance of the cloud detection algorithms. Additionally, the grouping of the different biomes is valuable, as they can then be evaluated individually.

To further compare the two datasets, the mean band value for each band in each scene in the two datasets was calculated, and the distributions can be compared (Fig. 3). This comparison shows discrepancies between the two datasets, where the SPARCS dataset contains fewer highly reflective pixels, caused primarily by the many cloud free scenes in the SPARCS dataset compared to the Biome dataset. It should be noted that for both the class and band distributions, the no-value pixels which surround the actual scene in the Biome dataset were not included. When the RS-Net model is trained on one dataset and evaluated on the other, it becomes evident that discrepancies in the evaluation are to be expected, due to the differences in both the class and band distributions. Furthermore, this will be exacerbated as the human analysts who annotated the datasets also disagree to some extent, as mentioned in the previous section.

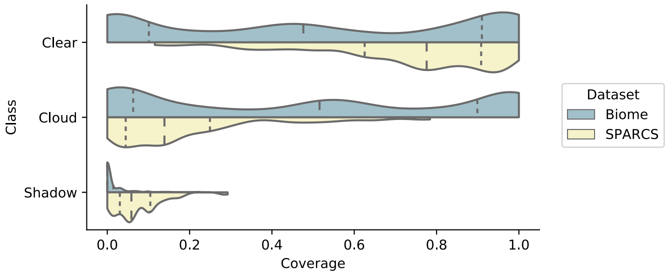


Fig. 2. Violin plot showing distribution of classes, with dotted lines showing the quartiles.

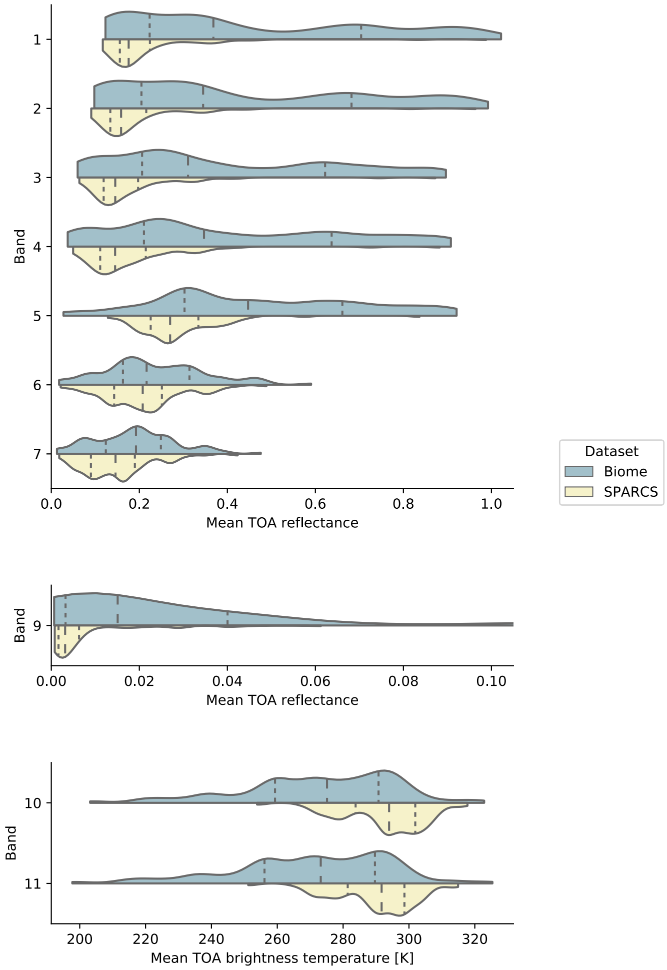
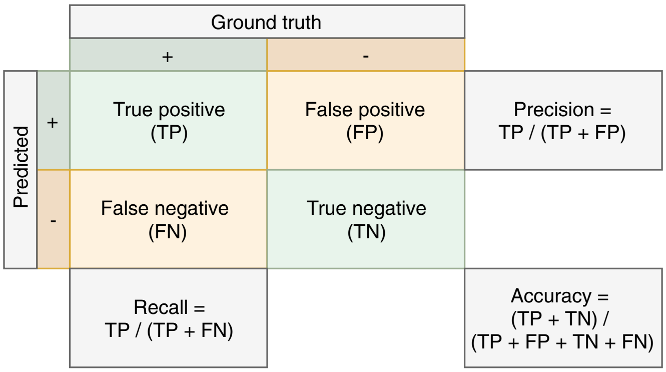


Fig. 3. Violin plot showing distributions of the Landsat 8 spectral bands in the used datasets, with dotted lines showing the quartiles. The bands from the OLI instrument cover the wavelengths from 0.435 μm (band 1) to 2.294 μm (band 7), and the thermal bands from the TIRS instrument cover the wavelenthgs 10.60 μm (band 10) to 12.51 μm (band 11). Band 9 is designated for detecting cirrus clouds. Band 8, the panchromatic band, is omitted in this work.

4.1. Evaluation metrics

The Landsat 8 datasets are evaluated quantitatively based on the accuracy, precision, and recall metrics, which are calculated based on a confusion matrix (Fig. 4) and over all valid pixels in a dataset simultaneously. The accuracy provides a good indication of the performance of the model, but does not account for imbalanced classes, and would lead to misinterpretations, for example, when evaluating a scene with barely any clouds present. Recall provides insight into the performance in capturing all true positives, thereby measuring how many of the cloud pixels were classified, disregarding the number of false positives. Precision gives insight into the amount of classified clouds, which were actually clouds. To find the optimum balance between the two, the F_1 score is calculated, as the harmonic mean between precision and recall:



		Ground truth		
		+	-	
Predicted	+	True positive (TP)	False positive (FP)	Precision = TP / (TP + FP)
	-	False negative (FN)	True negative (TN)	
		Recall = TP / (TP + FN)		Accuracy = (TP + TN) / (TP + FP + TN + FN)

Fig. 4. Confusion matrix and evaluation metrics.

$$F_1 = \frac{2}{Recall^{-1} + Precision^{-1}} \tag{6}$$

The output of RS-Net is a cloud confidence metric ranging from 0 to 1, therefore a threshold value must be determined for classification. In this paper, we fix the threshold value at 0.5 when evaluating the models, and subsequently investigate the results of basing the threshold value on a relationship between the precision and recall metrics, where a trade-off has to be made to determine how sensitive the classifier must be.

The evaluation of RS-Net is divided into two sections describing RS-Net validated on the SPARCS ground truth data, and RS-Net validated on the Biome ground truth data. In both cases, three different training datasets were used, each with five different band combinations. This results in a total of 30 separately trained models. The three training datasets were chosen to investigate the performance of RS-Net in three different scenarios. First, the RS-Net models are trained on the ground truth cloud masks from one dataset and tested on the ground truth cloud masks from the other (that is, trained on Biome and validated on SPARCS, and vice versa), to investigate the performance when the distribution of the training and test data are different. Second, the RS-Net models are trained and tested on the same dataset using image-based cross-validation, but the training data are the cloud masks produced with the Fmask algorithm, thereby using automatically generated training data. Third, the RS-Net models are trained and tested on the same dataset using image-based cross-validation, where the training data are the ground truth cloud masks, investigating the optimal performance of the RS-Net models. When applying image-based cross-validation, the SPARCS dataset was randomly divided into 5 folds, and the Biome dataset was divided into 2 folds with similar distributions regarding both the biomes and the cloud coverage. Several models were trained on different band combinations, and compared to the Fmask algorithm. It should be noted that the Python Fmask implementation (http://pythonfmask.org/en/latest/, n.d.) was used for all experiments in this paper. The results presented in the following section are based on accuracy and F_1 scores, however, all evaluation metrics can be found in Appendix A. The number of folds used for image-based cross-validation depends on the size of the dataset and a trade-off with training time, which increases with the number of folds applied. The size of the Biome dataset makes it feasible to only use 2 folds, where the smaller SPARCS dataset leads to improved performance with 5 folds.

5. Results and discussion

5.1. Landsat 8 - evaluation on SPARCS dataset

RS-Net models trained on the Biome ground truth and tested on SPARCS ground truth show similar accuracies but lower F_1 scores compared to the Fmask algorithm (Table 1). This issue is likely due to a combination of the fixed thresholding applied on the output of the RS-Net models and the dissimilarity between the class distributions of the datasets. In Appendix A, Table 6, the precision and recall metrics are included in the results. They show the RS-Net models being biased by the Biome datasets, leading to low recall values and high precision values, thereby leading to sub-optimal F_1 scores. This issue can be alleviated by fitting the threshold value to a subset of the test datasets, which will further discussed later in this section. When training on the cloud masks produced by the Fmask algorithm on the SPARCS dataset, and testing on the SPARCS ground truth (using 5-fold image-based cross-validation), all RS-Net models show similar accuracy compared to the Fmask algorithm. Surprisingly, the F_1 scores of the RS-Net models increase as the number of bands used decreases, with the single band model showing improved performance over the Fmask algorithm. This could be a consequence of the additional regularization introduced when discarding spectral information, and indicate that further improvements could be obtained through increased regularization, or by using a larger training dataset. The RS-Net models trained and tested on the SPARCS ground truth data using 5-fold image-based cross-validation show significantly increased performance, both regarding accuracy and F_1 scores. The number of spectral bands are of minor importance, showing the ability of RS-Net models in classifying clouds primarily based on the spatial patterns. These results are highly relevant for nano-satellites without the multi-spectral capabilities of the Landsat 8 satellite, which is exemplified by a single band RS-Net model successfully detecting clouds over a snowy region in a scene from the SPARCS dataset in Fig. 5.

Table 1
Evaluation results with RS-net models tested on the SPARCS dataset.

Model	Acc.	F_1
FMask	92.47	81.61
RS-Net trained on Biome ground truth and tested on SPARCS ground truth		
B = ALL	92.53	78.35
B = ALL-NT	93.26	80.62
B = RGBI	92.53	76.99
B = RGB	92.38	78.50
B = G	91.62	76.37
RS-Net trained on SPARCS Fmask and tested on SPARCS ground truth (5-fold CV)		
B = ALL	92.48	79.34
B = ALL-NT	92.00	80.19
B = RGBI	92.81	80.22
B = RGB	92.73	80.93
B = G	93.30	82.80
RS-Net trained on SPARCS ground truth and tested on SPARCS ground truth (5-fold CV)		
B = ALL	94.54	85.59
B = ALL-NT	**95.60**	**88.52**
B = RGBI	94.85	86.33
B = RGB	94.86	86.41
B = G	94.69	85.88

The bold text denotes the best performing method.

To further investigate the performance of the RS-Net models, the predicted cloud coverage was plotted against the ground truth cloud coverage (Fig. 6). The RS-Net models show significant improvements over the Fmask algorithm, with comparable results among models employing various spectral band combinations. It should be noted that the limitations of the SPARCS dataset become transparent here, as most of the data points are in the lower left quadrant of the scatter-plots (that is, no scenes with complete cloud coverage are evaluated).

The output of an RS-Net model is a cloud confidence metric, therefore the threshold value can be determined based on a trade-off between the precision and recall metrics, instead of the fixed value of 0.5 we used in this paper. More precisely, this trade-off is between the importance of being correct when predicting a cloud pixel (few false positives) and the importance of successfully predicting all cloud pixels (few false negatives). This trade-off is investigated through the relationship between the threshold values and the precision/recall, accuracy, and F_1 values (Fig. 7). The threshold value affects the

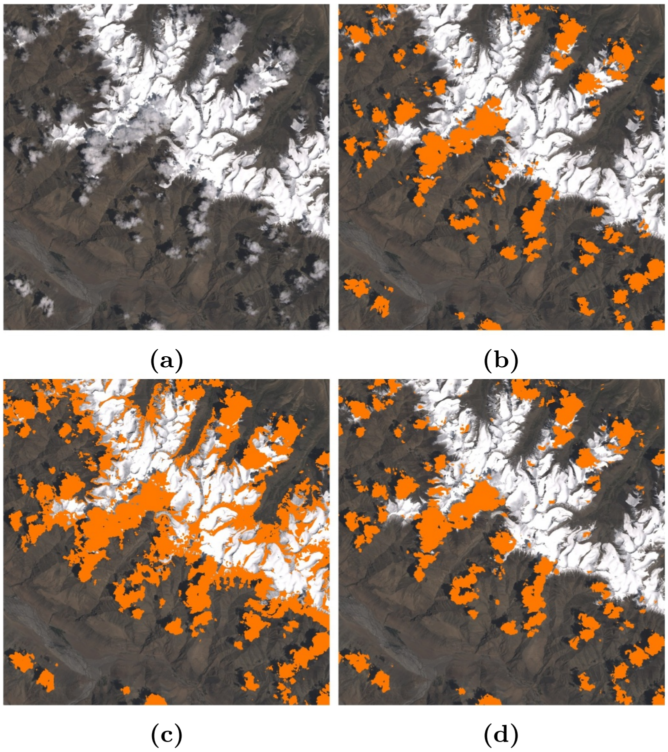


Fig. 5. Example of a SPARCS scene (LC81480352013195LGN00_32) classified using Python Fmask and RS-Net, showing (a) RGB scene, (b) ground truth, (c) Fmask prediction and (d) RS-Net single band prediction.

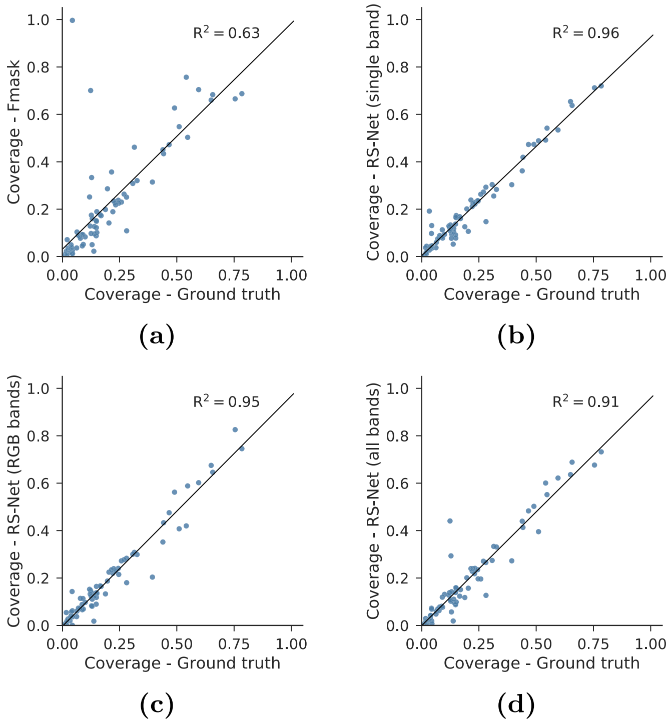


Fig. 6. The ground truth cloud coverage of the SPARCS dataset plotted against the predicted cloud coverage for (a) the Fmask algorithm, (b) RS-Net using a single band, (c) RS-Net using RGB bands, and (d) RS-Net using all bands.

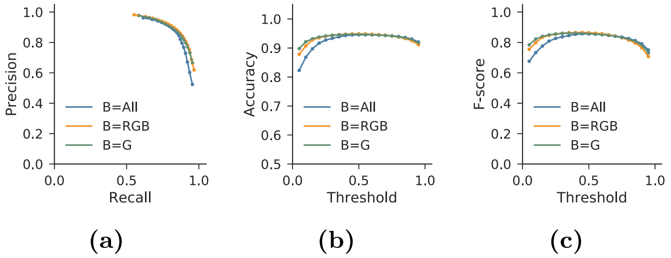


Fig. 7. Evaluation of different models on the SPARCS dataset, showing the models trained on the ground truth with (a) precision-recall curve, (b) accuracy vs. threshold curve, and (c) F-score vs. threshold curve.

Table 2
Evaluation results with RS-Net evaluated on the Biome dataset. Three of the eights biomes are presented, alongside the evaluation metrics of the entire dataset.

Model		Shrubl.	Snow/ Ice	Urban	Total
CFmask (Foga et al., 2017)	Acc.	87.30	64.09	92.05	88.48[a]
Fmask	Acc.	87.74	48.02	87.78	84.75
	F_1	87.04	53.93	86.71	85.03
RS-Net trained on SPARCS ground truth and tested on Biome ground truth					
B = All	Acc.	93.24	62.88	95.55	90.96
	F_1	93.20	62.92	95.35	91.02
B = All-NT	Acc.	94.03	68.24	95.64	91.59
	F_1	94.00	65.00	95.38	91.52
B = RGBI	Acc.	93.16	76.11	94.14	89.25
	F_1	93.02	69.21	93.63	88.77
B = RGB	Acc.	92.72	37.95	97.02	84.78
	F_1	92.45	53.55	96.81	85.35
B = G	Acc.	86.66	57.47	93.34	83.86
	F_1	85.39	61.12	92.68	83.07
RS-Net trained on Biome Fmask and tested on Biome ground truth (2-fold CV)					
B = All	Acc.	91.31	48.87	94.23	85.43
	F_1	91.36	55.35	94.03	86.10
B = All-NT	Acc.	93.45	51.42	95.81	86.65
	F_1	93.35	57.97	95.62	87.23
B = RGBI	Acc.	92.16	51.02	94.59	87.08
	F_1	91.95	54.28	94.42	87.43
B = RGB	Acc.	92.56	60.19	91.61	87.10
	F_1	92.76	63.73	91.48	87.66
B = G	Acc.	89.89	50.10	95.22	86.48
	F_1	89.52	57.94	94.95	86.93
RS-Net trained on Biome ground truth and tested on Biome ground truth (2-fold CV)					
B = All	Acc.	93.76	**88.39**	96.31	**93.81**
	F_1	93.43	**84.86**	96.04	**93.42**
B = All-NT	Acc.	94.71	84.33	95.86	93.14
	F_1	94.56	78.84	95.60	92.75
B = RGBI	Acc.	**95.04**	70.84	**97.30**	92.10
	F_1	**94.88**	63.07	**97.10**	91.73
B = RGB	Acc.	92.55	83.63	93.99	92.66
	F_1	92.01	77.28	93.74	92.23
B = G	Acc.	89.88	81.34	95.82	90.21
	F_1	88.71	73.27	95.47	89.15

The bold text denotes the best performing method.

[a] Calculated as the avg. value of the clear, midcl., and cloudy acc.

performance, and the models have optima regarding accuracy and F_1 scores.

5.2. Landsat 8 - evaluation on biome dataset

RS-Net models trained on the SPARCS dataset and evaluated on the Biome dataset show improved performance over the Fmask algorithm, except when a single spectral band is used (Table 2), where the performance is slightly inferior. Inspecting Table 8 in Appendix A, we see how the RS-Net models have been biased towards the SPARCS dataset, leading to low precision and high recall values. Therefore, it is expected that the performance can be further improved by fitting the threshold value to a subset of the test dataset. The results produced by the Python Fmask implementation differ from the CFmask results obtained in (Foga et al., 2017), although the implementations should be the same (Table 2). This might be caused by the inclusion of the buffered regions in the CFmask results, which are omitted in the Fmask results here. When the RS-Net models are trained on the output of the Fmask algorithm, they show improved performance compared to the Fmask algorithm. Similar to the results presented in the preceding section, the performance improves when fewer spectral bands are included, with the model using the RGB bands exhibiting the highest performance. This further indicates that reducing the spectral information improves the regularization of the model, and that improved regularization techniques should be investigated. However, the important finding is that RS-Net models can be trained to show improved performance over the algorithms used to produce the training data.

When the RS-Net models are trained and tested on the Biome ground truth data through 2-fold image-based cross-validation, the performance is significantly improved over the Fmask algorithm. In particular, the performance over snow/ice biomes is improved, where the accuracy and F_1 score are increased from 48.02 and 53.93 to 88.39 and 84.86 respectively for the model employing all spectral bands. Decreasing the spectral information has surprisingly low impact on the performance, where the accuracy and F_1 score of the entire Biome dataset decrease from 93.81 and 93.42 respectively for the all bands model to 90.21 and 89.15 respectively for the single band model. Thereby, the RGB and single band models show promising results for the use of RS-Net in satellites with limited multi-spectral capabilities, which is exemplified by the single band RS-Net model successfully detecting clouds over a difficult scene in Fig. 8.

The performance investigated by scatter-plotting the predicted cloud coverage against the ground truth cloud coverage show improved performance compared to the Fmask algorithm, particularly over snow/ice biomes (Fig. 9). The Fmask algorithm results in an R^2 value of 0.59, where the RGB and the all-band RS-Net models show similar performance, both with an R^2 value of 0.93. Compared to the evaluation on the SPARCS dataset (Fig. 6), the Biome dataset contains far more scenes completely covered in clouds. This can be seen by data-points being present in the upper right quadrant in the scatter-plots (Fig. 9), thereby resulting in a more reliable evaluation of the cloud detection algorithms.

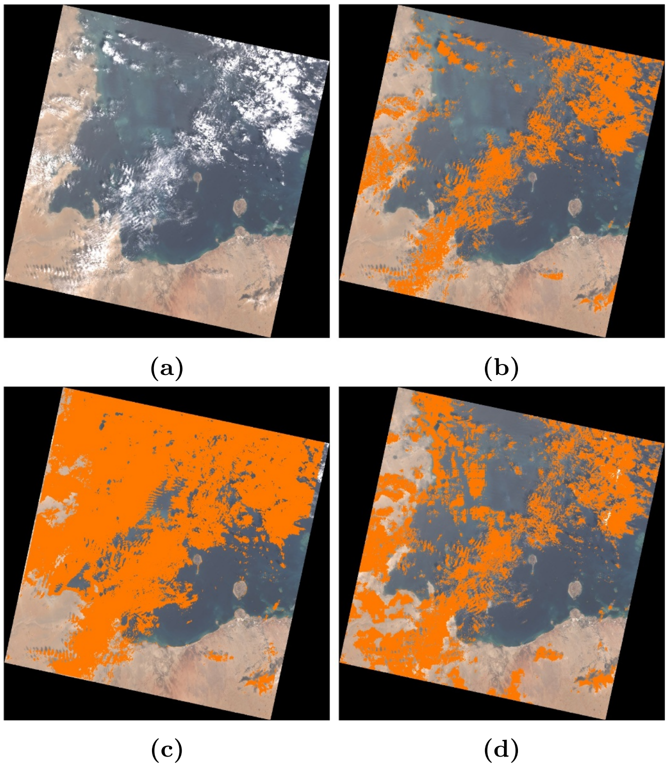
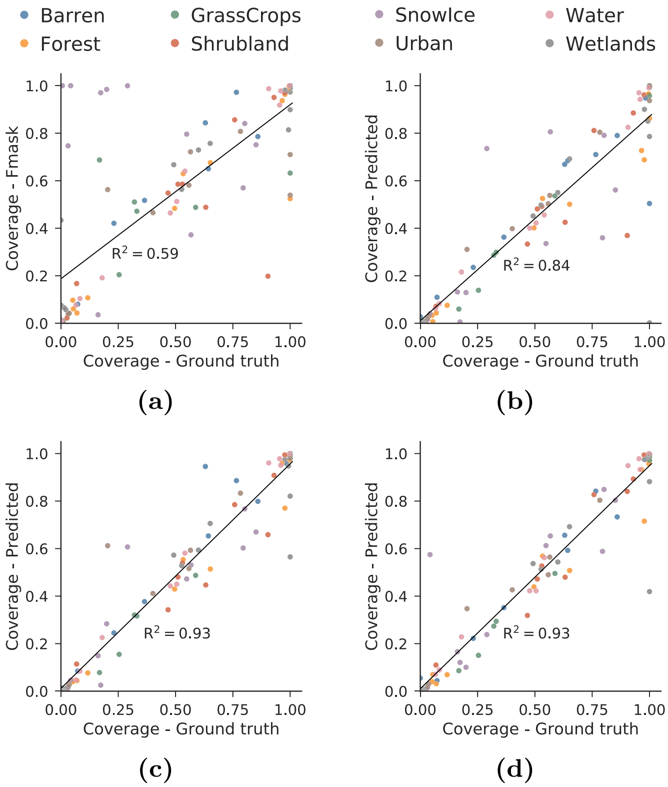

(a) (b) (c) (d)

Fig. 8. Example of a Biome scene (LC80650182013237LGN00) classified using Python Fmask and RS-Net, showing (a) RGB scene, (b) ground truth, (c) Fmask prediction, and (d) RS-Net single band prediction.



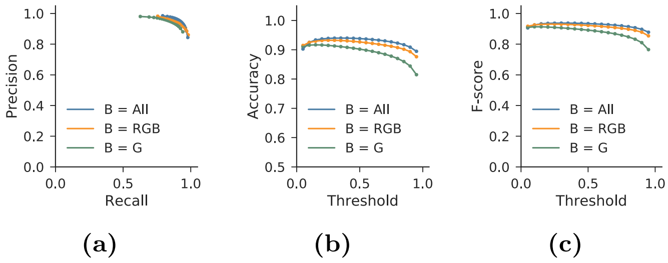
(a) (b) (c) (d)

Fig. 9. The ground truth cloud coverage of the Biome dataset plotted against the predicted cloud coverage for (a) the Fmask algorithm, (b) RS-Net using a single band, (c) RS-Net using RGB bands, and (d) RS-Net using all bands.

The relationship between the threshold value and the precision/recall relationship, the accuracy, and the F_1 scores is investigated (Fig. 10). The single band model benefits significantly from a low threshold, and the fixed value of 0.5 used in this paper can be tuned to improve the performance.

5.3. Timing analysis

The processing time of the RS-Net models vary depending on the number of spectral bands used, with the prediction time decreasing from 18.0 ± 2.4 s for the all-band model to 4.9 ± 0.4 s for the single



(a) (b) (c)

Fig. 10. Evaluation of different models on the Biome dataset, showing (a) precision-recall curve, (b) accuracy vs. threshold curve, and (c) F-score vs. threshold curve.

Table 3
Per product processing time of RS-Net models measured over all 96 products in the Biome dataset.

Model	Load	Predict	Save	Total
B = ALL	7.3 ± 0.9 s	18.0 ± 2.4 s	8.7 ± 1.5 s	34.0 ± 4.5 s
B = ALL-NT	6.1 ± 0.7 s	14.6 ± 1.9 s	8.1 ± 1.2 s	28.8 ± 3.8 s
B = RGBI	4.5 ± 0.6 s	8.8 ± 0.8 s	8.7 ± 1.6 s	22.0 ± 3.0 s
B = RGB,	4.2 ± 0.5 s	7.5 ± 0.6 s	9.8 ± 2.4 s	21.5 ± 3.5 s
B = G	3.8 ± 0.5 s	4.9 ± 0.4 s	8.0 ± 1.1 s	16.6 ± 2.0 s

band model (Table 3). It should be noted that the outputs are saved as PNG files to save disk space, thereby leading to increased save times due to the image compression required, and that the results presented here are based on a single-threaded implementation. A multi-threaded implementation which concurrently load/save and process data would reduce the total processing time to be close to the prediction time.

Thresholding the prediction was measured to take 0.08 ± 0.01 s per product (measured over all 96 products in the Biome dataset). The timing analysis was done with an Intel Core i7-6800K, 64 GB DDR4 RAM, 1 TB Samsung 960 Pro SSD, and 2 x Nvidia GTX 1080Ti, using Keras 2.1.6 and TensorFlow 1.9.0 (with CUDA 9.0 and CUDnn 7).

5.4. RS-Net evaluation

RS-Net was evaluated on the Landsat 8 SPARCS and Biome datasets, though most emphasis will be put on the results from the Biome dataset. This is due to two specific reasons; (i), the Biome dataset is more than hundred times larger than the SPARCS dataset, and (ii), it contains scenes completely covered in clouds, which is a deficit of the SPARCS dataset. The single band cloud detection models evaluated here showed that the spatial patterns are of high importance in cloud detection algorithms, and that the RS-Net models successfully include the spatial patterns in the cloud detection, thereby leading to improved performance over traditional methods. This is of high importance when performing cloud detection based on satellite imagery from satellites without the multi-spectral capabilities of larger satellites. In addition to high performance using few spectral bands, the performance when classifying clouds over snow/ice was significantly improved. It is important to note, however, that the RS-Net model does not necessarily invalidate previous cloud detection methods. On the contrary, it acts as a supplement to them, which manages to avoid the corner-cases where other methods break down, such as over snow/ice biomes. Much a priori knowledge was collected in the design of the decision tree based methods (Foga et al., 2017), and a hybrid model based on the RS-Net prediction as input to the Fmask algorithm could improve the performance even further.

Deep learning models require large amounts of training data. Therefore, we investigated the performance of RS-Net models when they are trained on the output from the Fmask algorithm. Deep learning models have shown to be resilient to noisy training data, and deep neural networks are capable of generalizing even though there is a majority of incorrect labels in the training dataset (Rolnick et al., 2017). When we use the output of the Fmask algorithm on the Biome dataset as training data for the RS-Net models, the performance is improved compared to the Fmask algorithm. This could potentially be improved further by a recursive implementation, where models are trained and used to produce improved training data (Triguero et al., 2015). It does however require the training dataset to be large enough, and the model to be regularized sufficiently. The results presented here showed that decreasing the spectral information improved the gain in performance compared to the Fmask algorithm, which is likely due to the network learning the errors in the training dataset when the spectral information is not constricted. By reducing the spectral information, the learning capabilities of RS-Net is reduced, thereby acting as additional regularization. This indicates that further improvements can be expected for training on noisy data, and that more sophisticated regularization strategies should be investigated to maximize the gain in performance when employing all spectral bands.

The total processing time of 34.0 ± 4.5 s per Landsat 8 product with the largest RS-Net model is satisfactory for production environments, particularly due to the straightforward improvements which can be obtained through a multi-threaded implementation. Additionally, hardware and software for deep learning is rapidly improving.

We modified the original U-net architecture by employing batch normalization, inserting a clipping layer, halving the depth of the feature maps, and by changing the activation function. These changes resulted in reduced requirements to pre-processing, faster training and inference time, and improved performance. The improvements in performance over traditional methods are primarily due to better inclusion of the spatial patterns in the classification task, exemplified by performing cloud detection on a single spectral band. That is, the texture of the cloud is more important for the classification than the spectral signature, demonstrated by the high performance of the single-band model. Interpretability of deep learning models is currently subject to much research, including both visualizations of the learned filters or the creation of graphs designed to explain the classifications (Zhang and Zhu, 2018). In the context of cloud detection, this can be used, for instance, to further investigate the importance of spatial and spectral information for successful cloud classification. The construction of explanatory graphs and decision trees to complement a classification can be valuable both for the interpretability of the individual deep learning models and for research in investigating, understanding, and improving the architectures (Zhang and Zhu, 2018). Although these methods are outside the scope of this paper, they are important for future research. RS-Net can act as a facilitator for future scientific research, where reliable cloud detection is necessary, with planned extensions including both methods for improving the performance and for interpretability. Complementing the classifications with, e.g., an auto-generated document providing information on the analysis of the scene, including the reasoning for the classification output, can strengthen the trust in the method. Furthermore, deep learning models are increasingly used in the scientific domains (Montavon et al., 2018), where methods for interpretability has been used to gain insights in physical, chemical, and biological systems (Alipanahi et al., 2015; Schütt et al., 2017; Sturm et al., 2016). Deep neural networks for remote sensing classification purposes could potentially be combined with advanced methods for interpretability to investigate physical properties of the sensed areas, rather than merely acting as classification algorithms.

6. Conclusion

The Landsat 8 programme has increased the usage of satellite imagery significantly due to its free data policy. In addition, ESA complements this open satellite data repository with its Sentinel satellites, and it can only be expected that satellite data analyses will become even more widespread in coming years. One of the first preprocessing steps when employing optical satellite data is to do cloud-masking, where cloud pixels are identified and omitted in further analysis. The importance of this task is becoming critical due to the increasing amounts of data being used for automated analytics, where there is no human assessment of the scenes before they are used. Several methods exist for cloud detection, however, they are mainly based on the spectral signatures of the individual pixels, and does not take full advantage of incorporating the spatial patterns into the classification. RS-Net is a deep learning algorithm based on the U-net architecture (Ronneberger et al., 2015), which aims to detect clouds using a combination of spatial and spectral patterns. It improves the accuracy over the Fmask algorithm, and shows high performance using only the RGB and RGBI bands. This is of particular interest for low-cost nanosatellites, without the multi-spectral capabilities of large satellites, like the Landsat 8 satellite. Additionally, it is shown how the model is resistant to noisy training data. This is illustrated by training RS-Net models on the output of the Fmask algorithm, which show improved performance compared to the Fmask algorithm. Finally, the average classification time of a product in the Biome dataset (96 full Landsat 8 scenes) was 18.0 ± 2.4 s for the largest RS-Net model on a prosumer PC.

Although the current RS-Net models perform well, there are several directions for further improvements. A digital elevation model could be included by concatenating it to the input as an additional spectral band, and ensemble methods can be used to obtain improved prediction performance by averaging the output from several models. New architectures could be investigated, either based on U-net, or entirely new ones could be designed. Hyperparameter optimization, including optimizing the network architecture, is currently subject to much research, and it is expected that this will be significantly improved through global optimization strategies in coming years. Finally, the a priori knowledge gathered in the last decades to create cloud detection algorithms could be incorporated into the model as a post-processing step.

Future work will focus on extending RS-Net to perform multi-class prediction, include methods for interpretability regarding the classification decisions, and to incorporate the temporal patterns into the model. Incorporating the temporal patterns are of particularly high importance in remote sensing, where a building remains static over time, different types of vegetation follow different slow and predictable cycles, and clouds follow a very fast and more random cycle. Multi-temporal semantic segmentation methods based on deep learning is still computationally expensive. However, current research shows promising results (Jampani et al., 2017; Shelhamer et al., 2016), and it is expected that methods which fully incorporate the spectral, spatial, and temporal dimensions will be developed in the near future.

Acknowledgments

This work was funded by Innovation Fund Denmark as part of the Future Cropping project (J. nr. 5107-00002B).

Appendix A

Table 4 provides an overview of the essential software packages used in the paper. Tables 5 and 7 provide the hyperparameters found by through random search when evaluating on the SPARCS and Biome dataset respectively. The input image size was fixed at 256 × 256 pixels, with a batch size of 40 for models evaluated on the Biome dataset and 16 for models evaluated on the SPARCS dataset. The GPU memory is the limiting factor with respect to batch size, and the different batch sizes was due to two different PCs being used for running experiments, one with 2 x Nvidia GTX 1080ti when evaluating on the Biome dataset and one with an Nvidia GTX 1080 when evaluating on the SPARCS dataset. Tables 6 and 8 provide all results when evaluated on the SPARCS and Biome dataset respectively.

Table 4
Software setup.

Name	Ver.	Description
Ubuntu	16.04	Linux operating system.
Python	3.6	Programming language.
Keras	2.1.6	High-level API used for TensorFlow.
TensorFlow	1.9.0	Deep learning framework by Google.
CUDA	9.0	Platform for GPU based processing (used by TensorFlow).
CUDnn	7	Library for CUDA for deep neural networks (used by TensorFlow/CUDA).
Python Fmask	0.4.5	Implementation of the Fmask algorithm.

Table 5
Hyperparameters for final models when the SPARCS dataset was used for evaluation.

Model	LR ($\times 10^{-3}$)	Dropout	L2 ($\times 10^{-3}$)	Epochs
RS-Net trained on Biome ground truth and tested on SPARCS ground truth				
B = ALL	0.88	0.00	0.73	14
B = ALL-NT	0.44	0.39	0.87	39
B = RGBI	0.78	0.38	0.15	45
B = RGB,	0.82	0.01	0.15	62
B = G	0.23	0.00	0.60	8
RS-Net trained on SPARCS Fmask and tested on SPARCS ground truth (5-fold CV)				
B = ALL	0.43	0.27	2.56	11
B = ALL-NT	0.38	0.00	0.71	133
B = RGBI	0.11	0.00	0.67	11
B = RGB,	0.20	0.00	0.78	46
B = G	0.16	0.00	0.06	74
RS-Net trained on SPARCS ground truth and tested on SPARCS ground truth (5-fold CV)				
B = ALL	0.45	0.04	0.23	44
B = ALL-NT	0.21	0.00	0.58	53
B = RGBI	0.44	0.44	0.80	84
B = RGB,	0.42	0.00	0.28	66
B = G	0.31	0.41	0.26	39

Table 6
Evaluation results with RS-net evaluated on the SPARCS dataset.

Model	Acc.	Prec.	Rec.	F_1
FMask	92.47	77.47	86.21	81.61
RS-Net trained on Biome ground truth and tested on SPARCS ground truth				
B = ALL	92.53	88.57	70.53	78.35
B = ALL-NT	93.26	91.04	72.34	80.62
B = RGBI	92.53	95.35	64.56	76.99
B = RGB	92.38	86.54	71.83	78.50
B = G	91.62	84.17	69.90	76.37
RS-Net trained on SPARCS Fmask and tested on SPARCS ground truth (5-fold CV)				
B = ALL	92.48	84.87	74.49	79.34
B = ALL-NT	92.00	77.16	83.46	80.19
B = RGBI	92.81	85.93	75.22	80.22
B = RGB	92.73	82.33	79.58	80.93
B = G	93.30	82.32	83.29	82.80
RS-Net trained on SPARCS ground truth and tested on SPARCS ground truth (5-fold CV)				
B = ALL	94.54	87.62	83.66	85.59
B = ALL-NT	95.60	89.47	87.58	88.52
B = RGBI	94.85	88.92	83.89	86.33
B = RGB	94.86	88.58	84.34	86.41
B = G	94.69	88.51	83.40	85.88

Table 7
Hyperparameters for final models when the Biome dataset was used for evaluation.

Model	LR	Dropout	L2	Epochs
RS-Net trained on SPARCS ground truth and tested on Biome ground truth				
B = ALL	0.22	0.17	0.18	25
B = ALL-NT	0.39	0.38	0.46	76
B = RGBI	0.21	0.00	0.47	41
B = RGB,	0.76	0.00	0.73	3
B = G	0.23	0.00	0.65	38
RS-Net trained on Biome Fmask and tested on Biome ground truth (2-fold CV)				
B = ALL	0.57	0.36	6.89	30
B = ALL-NT	0.40	0.22	8.95	36
B = RGBI	0.11	0.01	0.95	48
B = RGB,	0.41	0.00	0.66	53
B = G	0.13	0.04	0.95	60
RS-Net trained on Biome ground truth and tested on Biome ground truth (2-fold CV)				
B = ALL	0.70	0.00	0.11	21
B = ALL-NT	0.97	0.00	0.99	42
B = RGBI	0.38	0.00	0.88	31
B = RGB,	0.16	0.00	0.99	117
B = G	0.39	0.00	0.20	52

Table 8
Evaluation results with RS-net evaluated on the Biome dataset.

Model		Barren	Forest	Grass/Cr.	Shrubl.	Snow/Ice	Urban	Water	Wetl.	Clear[a]	MidCl.	Cloudy	Total
CFmask (Foga et al., 2017)	Acc.	92.39	93.83	94.07	87.30	64.09	92.05	86.36	92.38	92.91	86.79	85.74	88.48[b]
Fmask	Acc.	91.23	92.58	87.82	87.74	48.02	87.78	94.76	88.11	86.33	82.99	84.94	84.75
	Prec.	85.96	96.33	85.30	90.04	40.26	87.44	92.83	84.45	21.02	76.51	94.06	80.34
	Rec.	96.93	90.64	89.85	84.24	81.66	85.99	96.38	94.56	77.57	93.88	88.99	90.31
	F_1	91.12	93.40	87.52	87.04	53.93	86.71	94.57	89.22	33.07	84.31	91.46	85.03
RS-Net trained on SPARCS ground truth and tested on Biome ground truth													
B = ALL	Acc.	93.85	94.02	95.74	93.24	62.88	95.55	95.81	96.64	92.55	89.12	91.23	90.96
	Prec.	89.13	97.17	94.79	91.69	50.11	92.40	93.60	95.06	35.48	84.64	94.26	86.98
	Rec.	98.78	92.37	96.33	94.76	84.54	98.50	97.85	98.68	86.90	94.86	96.17	95.45
	F_1	93.71	94.71	95.55	93.20	62.92	95.35	95.68	96.84	50.39	89.46	95.21	91.02
B = ALL-NT	Acc.	93.19	94.69	95.26	94.03	68.24	95.64	96.09	95.66	93.03	90.39	91.38	91.59
	Prec.	91.60	96.68	93.68	92.46	55.14	93.87	93.49	94.25	37.20	88.33	94.64	88.72
	Rec.	93.93	94.06	96.53	95.59	79.15	96.93	98.63	97.63	87.35	92.48	95.91	94.49
	F_1	92.75	95.35	95.08	94.00	65.00	95.38	95.99	95.91	52.18	90.36	95.27	91.52
B = RGBI	Acc.	91.74	90.41	89.53	93.16	76.11	94.14	90.23	88.80	94.43	87.69	85.67	89.25
	Prec.	87.54	93.51	93.30	92.79	66.58	94.26	92.38	88.71	42.79	84.11	97.04	89.04
	Rec.	95.84	89.66	84.00	93.24	72.06	93.01	86.52	89.92	82.37	92.13	86.83	88.49
	F_1	91.50	91.55	88.41	93.02	69.21	93.63	89.35	89.31	56.32	87.94	91.65	88.77

(continued on next page)

Table 8 (*continued*)

Model		Barren	Forest	Grass/Cr.	Shrubl.	Snow/Ice	Urban	Water	Wetl.	Clear[a]	MidCl.	Cloudy	Total
B = RGB	Acc.	85.56	91.31	94.65	92.72	37.95	97.02	91.46	87.76	79.55	82.92	91.84	84.78
	Prec.	82.83	93.75	97.91	93.67	37.13	95.98	89.51	84.60	14.60	81.12	93.90	79.32
	Rec.	86.90	91.06	90.68	91.27	96.00	97.65	92.86	93.51	76.20	84.61	97.31	92.37
	F_1	84.82	92.39	94.16	92.45	53.55	96.81	91.16	88.83	24.51	82.83	95.58	85.35
B = G	Acc.	85.51	85.30	86.49	86.66	57.47	93.34	87.81	88.40	91.11	83.62	76.90	83.86
	Prec.	88.15	95.73	93.40	91.92	46.34	94.37	89.45	93.40	30.11	79.64	94.45	83.62
	Rec.	79.46	78.10	77.01	79.72	89.72	91.06	84.20	83.62	78.78	89.14	79.14	82.52
	F_1	83.58	86.02	84.42	85.39	61.12	92.68	86.75	88.24	43.57	84.12	86.12	83.07
RS-Net trained on Biome Fmask and tested on Biome ground truth (2-fold CV)													
B = ALL	Acc.	87.37	91.24	88.30	91.31	48.87	94.23	93.79	88.49	84.24	82.49	89.57	85.43
	Prec.	79.80	95.34	82.42	88.82	41.02	90.32	93.04	83.11	19.41	76.05	93.68	79.39
	Rec.	97.48	89.24	95.81	94.06	85.06	98.07	93.91	97.75	83.10	93.47	94.88	94.05
	F_1	87.75	92.19	88.61	91.36	55.35	94.03	93.47	89.84	31.48	83.86	94.28	86.10
B = ALL-NT	Acc.	91.76	87.11	90.63	93.45	51.42	95.81	94.63	88.59	84.19	84.56	91.20	86.65
	Prec.	87.41	86.56	86.36	92.62	42.77	92.59	92.82	83.43	19.75	78.66	94.20	80.61
	Rec.	96.10	92.02	95.32	94.10	89.94	98.86	96.11	97.43	85.84	93.69	96.21	95.05
	F_1	91.55	89.21	90.62	93.35	57.97	95.62	94.44	89.89	32.11	85.52	95.20	87.23
B = RGBI	Acc.	87.33	90.60	92.68	92.16	51.02	94.59	94.65	93.70	88.08	83.91	89.25	87.08
	Prec.	82.19	90.05	87.61	92.31	41.61	90.50	93.04	92.26	23.35	77.85	93.67	81.99
	Rec.	92.81	94.17	98.51	91.59	78.03	98.70	95.88	95.95	76.05	93.58	94.52	93.64
	F_1	87.18	92.06	92.74	91.95	54.28	94.42	94.44	94.07	35.73	85.00	94.09	87.43
B = RGB	Acc.	82.81	92.40	90.11	92.56	60.19	91.61	93.59	93.58	89.94	81.95	89.43	87.10
	Prec.	76.25	93.99	84.33	88.43	48.24	86.44	89.92	91.63	28.02	74.33	92.84	81.02
	Rec.	91.45	92.80	97.25	97.54	93.88	97.14	97.38	96.49	83.46	96.13	95.71	95.48
	F_1	83.16	93.39	90.33	92.76	63.73	91.48	93.50	93.99	41.96	83.84	94.25	87.66
B = G	Acc.	86.24	92.71	94.66	89.89	50.10	95.22	94.36	88.78	86.33	84.89	88.22	86.48
	Prec.	79.70	94.89	92.01	90.78	42.23	92.89	91.66	87.37	21.60	78.59	93.28	81.06
	Rec.	94.42	92.38	97.20	88.29	92.26	97.11	96.93	91.70	81.26	94.79	93.75	93.73
	F_1	86.44	93.62	94.53	89.52	57.94	94.95	94.22	89.49	34.12	85.94	93.51	86.93
RS-Net trained on Biome ground truth and tested on Biome ground truth (2-fold CV)													
B = ALL	Acc.	95.15	93.36	95.86	93.76	88.39	96.31	95.74	91.94	97.96	91.35	92.16	93.81
	Prec.	95.40	98.58	99.27	96.30	82.52	95.35	95.57	97.81	80.63	94.45	96.46	95.37
	Rec.	94.10	89.83	91.96	90.73	87.34	96.75	95.43	86.46	69.98	87.35	94.82	91.54
	F_1	94.74	94.00	95.47	93.43	84.86	96.04	95.50	91.78	74.93	90.76	95.63	93.42
B = ALL-NT	Acc.	95.46	93.79	96.46	94.71	84.33	95.86	95.39	89.20	95.91	91.07	92.47	93.14
	Prec.	95.60	96.48	97.80	95.12	79.35	94.18	94.51	96.01	52.14	92.52	97.92	94.07
	Rec.	94.57	92.66	94.68	94.00	78.34	97.07	95.84	82.68	75.33	88.84	93.67	91.48
	F_1	95.08	94.53	96.22	94.56	78.84	95.60	95.17	88.85	61.63	90.64	95.75	92.75
B = RGBI	Acc.	95.21	93.93	96.24	95.04	70.84	97.30	94.77	93.56	93.24	88.66	94.42	92.10
	Prec.	97.77	96.15	98.46	95.76	59.70	96.59	95.03	93.83	34.98	91.32	97.68	92.15
	Rec.	91.77	93.24	93.55	94.02	66.84	97.61	93.87	93.80	64.36	84.76	96.12	91.31
	F_1	94.68	94.67	95.94	94.88	63.07	97.10	94.45	93.82	45.32	87.92	96.90	91.73
B = RGB	Acc.	93.10	93.97	95.87	92.55	83.63	93.99	95.97	92.21	97.63	87.61	92.78	92.66
	Prec.	89.76	98.51	98.39	96.69	80.05	90.58	95.29	96.69	79.33	87.04	98.00	93.69
	Rec.	96.11	90.96	92.84	87.76	74.69	97.14	96.26	88.05	61.73	87.60	93.95	90.82
	F_1	92.82	94.59	95.53	92.01	77.28	93.74	95.77	92.17	69.43	87.32	95.93	92.23
B = G	Acc.	89.03	90.00	95.43	89.88	81.34	95.82	94.57	85.68	97.45	87.81	85.42	90.21
	Prec.	92.06	99.29	98.54	97.57	78.54	95.91	96.69	98.07	79.39	90.38	98.50	95.14
	Rec.	83.57	83.32	91.73	81.33	68.66	95.04	91.69	73.94	56.00	83.89	85.19	83.87
	F_1	87.61	90.61	95.01	88.71	73.27	95.47	94.12	84.31	65.67	87.02	91.37	89.15

[a] The precision, recall, and F_1 scores are unreliable when evaluating clear products, as these metrics are calculated based on cloudy pixels.

[b] Calculated as the avg. value of the clear, midcl., and cloudy acc.

References

Alipanahi, B., Delong, A., Weirauch, M.T., Frey, B.J., 2015. Predicting the sequence specificities of dna-and rna-binding proteins by deep learning. Nat. Biotechnol. 33 (8), 831.

Badrinarayanan, V., Kendall, A., Cipolla, R., 2017. Segnet: a deep convolutional encoder-decoder architecture for image segmentation. IEEE Trans. Pattern Anal. Mach. Intell. 39 (12), 2481–2495.

Chen, Y., Fan, R., Bilal, M., Yang, X., Wang, J., Li, W., 2018. Multilevel cloud detection for high-resolution remote sensing imagery using multiple convolutional neural networks. ISPRS Int. J. Geo-Information 7 (5), 181.

Choromanska, A., Henaff, M., Mathieu, M., Arous, G.B., LeCun, Y., 2015. The loss surfaces of multilayer networks. In: Artificial Intelligence and Statistics, pp. 192–204.

Clevert, D.-A., Unterthiner, T., Hochreiter, S., 2015. Fast and Accurate Deep Network Learning by Exponential Linear Units (Elus), arXiv Preprint arXiv:1511.07289. (NaN–NaN).

Drönner, J., Korfhage, N., Egli, S., Mühling, M., Thies, B., Bendix, J., Freisleben, B., Seeger, B., 2018. Fast cloud segmentation using convolutional neural networks. Remote Sens. 10 (11), 1782.

Ferencz, C., Bognar, P., Lichtenberger, J., Hamar, D., Tarcsai, G., Timár, G., Molnár, G., Pásztor, S., Steinbach, P., Székely, B., et al., 2004. Crop yield estimation by satellite remote sensing. Int. J. Remote Sens. 25 (20), 4113–4149.

Foga, S., Scaramuzza, P.L., Guo, S., Zhu, Z., Dilley, R.D., Beckmann, T., Schmidt, G.L., Dwyer, J.L., Hughes, M.J., Laue, B., 2017. Cloud detection algorithm comparison and validation for operational landsat data products. Remote Sens. Environ. 194, 379–390.

Glorot, X., Bengio, Y., 2010. Understanding the difficulty of training deep feedforward neural networks. In: Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, pp. 249–256.

Hong, D., Yokoya, N., Ge, N., Chanussot, J., Zhu, X.X., 2019. Learnable manifold alignment (lema): a semi-supervised cross-modality learning framework for land cover and land use classification. ISPRS J. Photogramm. Remote Sens. 147, 193–205.

http://pythonfmask.org/en/latest/, Accessed date: 22 August 2018.

https://github.com/JacobJeppesen/RS-Net, Accessed date: 16 January 2019.

https://landsat.usgs.gov/what-are-band-designations-landsat-satellites, Accessed date: 2 July 2018.

Hughes, M.J., Hayes, D.J., 2014. Automated detection of cloud and cloud shadow in single-date landsat imagery using neural networks and spatial post-processing. Remote Sens. 6 (6), 4907–4926.

Ioffe, S., Szegedy, C., 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: arXiv preprint arXiv:1502.03167, NaN–NaN.

Jampani, V., Gadde, R., Gehler, P.V., 2017. Video propagation networks. In: Proc. CVPR. vol. 6. pp. 7.

Joyce, K.E., Belliss, S.E., Samsonov, S.V., McNeill, S.J., Glassey, P.J., 2009. A review of the status of satellite remote sensing and image processing techniques for mapping natural hazards and disasters. Prog. Phys. Geogr. 33 (2), 183–207.

Karakizi, C., Karantzalos, K., Vakalopoulou, M., Antoniou, G., 2018. Detailed land cover mapping from multitemporal landsat-8 data of different cloud cover. Remote Sens. 10 (8), 1214.

Kingma, D.P., Ba, J., Adam, 2015. A method for stochastic optimization. In: International Conference on Learning Representations, (NaN–NaN).

Le Goff, M., Tourneret, J.-Y., Wendt, H., Ortner, M., Spigai, M., 2017. Deep learning for cloud detection. In: ICPRS (8th International Conference of Pattern Recognition Systems), pp. 1–6.

Li, R., Liu, W., Yang, L., Sun, S., Hu, W., Zhang, F., Li, W., 2018a. Deepunet: a deep fully convolutional network for pixel-level sea-land segmentation. IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. 11 (11), 3954–3962.

Li, X., Chen, S., Hu, X., Yang, J., 2018b. Understanding the Disharmony between Dropout and Batch Normalization by Variance Shift. arXiv preprint arXiv:1801.05134. (NaN–NaN).

Long, J., Shelhamer, E., Darrell, T., 2015. Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3431–3440.

Mateo-García, G., Gómez-Chova, L., Amorós-López, J., Muñoz-Marí, J., Camps-Valls, G., 2018. Multitemporal cloud masking in the google earth engine. Remote Sens. 10 (7), 1079.

Montavon, G., Samek, W., Müller, K.-R., 2018. Methods for interpreting and understanding deep neural networks. Digital Signal Process. 73, 1–15.

Ng, A.Y., 2004. Feature selection, l 1 vs. l 2 regularization, and rotational invariance. In: Proceedings of the Twenty-First International Conference on Machine Learning. ACM, pp. 78.

Prasad, A.K., Chai, L., Singh, R.P., Kafatos, M., 2006. Crop yield estimation model for Iowa using remote sensing and surface parameters. Int. J. Appl. Earth Obs. Geoinf. 8 (1), 26–33.

Reddi, S.J., Kale, S., Kumar, S., 2018. On the convergence of Adam and beyond. In: International Conference on Learning Representations, (NaN–NaN).

Rolnick, D., Veit, A., Belongie, S., Shavit, N., 2017. Deep Learning Is Robust to Massive Label Noise. arXiv preprint arXiv:1705.10694. (NaN–NaN).

Ronneberger, O., Fischer, P., Brox, T., 2015. U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. Springer, pp. 234–241.

Rumelhart, D.E., Hinton, G.E., Williams, R.J., 1986. Learning representations by back-propagating errors. nature 323 (6088), 533.

Sakamoto, T., Yokozawa, M., Toritani, H., Shibayama, M., Ishitsuka, N., Ohno, H., 2005. A crop phenology detection method using time-series modis data. Remote Sens. Environ. 96 (3–4), 366–374.

Scaramuzza, P.L., Bouchard, M.A., Dwyer, J.L., 2012. Development of the landsat data continuity mission cloud-cover assessment algorithms. IEEE Trans. Geosci. Remote Sens. 50 (4), 1140–1154.

Schütt, K.T., Arbabzadah, F., Chmiela, S., Müller, K.R., Tkatchenko, A., 2017. Quantum-chemical insights from deep tensor neural networks. Nat. Commun. 8, 13890.

Shelhamer, E., Rakelly, K., Hoffman, J., Darrell, T., 2016. Clockwork convnets for video semantic segmentation. In: European Conference on Computer Vision. Springer, pp. 852–868.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R., 2014. Dropout: a simple way to prevent neural networks from overfitting. J. Mach. Learn. Res. 15 (1), 1929–1958.

Sturm, I., Lapuschkin, S., Samek, W., Müller, K.-R., 2016. Interpretable deep neural networks for single-trial eeg classification. J. Neurosci. Methods 274, 141–145.

Telea, A., 2004. An image inpainting technique based on the fast marching method. J. Graph. Tools 9 (1), 23–34.

Tralli, D.M., Blom, R.G., Zlotnicki, V., Donnellan, A., Evans, D.L., 2005. Satellite remote sensing of earthquake, volcano, flood, landslide and coastal inundation hazards. ISPRS J. Photogramm. Remote Sens. 59 (4), 185–198.

Triguero, I., García, S., Herrera, F., 2015. Self-labeled techniques for semi-supervised learning: taxonomy, software and empirical study. Knowl. Inf. Syst. 42 (2), 245–284.

Verbesselt, J., Hyndman, R., Newnham, G., Culvenor, D., 2010. Detecting trend and seasonal changes in satellite image time series. Remote Sens. Environ. 114 (1), 106–115.

Voigt, S., Kemper, T., Riedlinger, T., Kiefl, R., Scholte, K., Mehl, H., 2007. Satellite image analysis for disaster and crisis-management support. IEEE Trans. Geosci. Remote Sens. 45 (6), 1520–1528.

Zhang, Q.-s., Zhu, S.-C., 2018. Visual interpretability for deep learning: a survey. Front. Inf. Technol. Electron. Eng. 19 (1), 27–39.

Zhang, Y., Rossow, W.B., Lacis, A.A., Oinas, V., Mishchenko, M.I., 2004. Calculation of radiative fluxes from the surface to top of atmosphere based on isccp and other global data sets: refinements of the radiative transfer model and the input data. J. Geophys. Res.-Atmos. 109, D19105 (NaN–NaN).

Zhang, Z., Liu, Q., Wang, Y., 2018. Road extraction by deep residual u-net. IEEE Geosci. Remote Sens. Lett. 15, 749–753.

Zhu, Z., 2017. Change detection using landsat time series: a review of frequencies, preprocessing, algorithms, and applications. ISPRS J. Photogramm. Remote Sens. 130, 370–384.

Zhu, Z., Woodcock, C.E., 2012. Object-based cloud and cloud shadow detection in landsat imagery. Remote Sens. Environ. 118, 83–94.

Zhu, Z., Wang, S., Woodcock, C.E., 2015. Improvement and expansion of the fmask algorithm: cloud, cloud shadow, and snow detection for landsats 4–7, 8, and sentinel 2 images. Remote Sens. Environ. 159, 269–277.

Zhu, X.X., Tuia, D., Mou, L., Xia, G.-S., Zhang, L., Xu, F., Fraundorfer, F., 2017. Deep learning in remote sensing: a comprehensive review and list of resources. IEEE Geosci. Remote Sens. Mag. 5 (4), 8–36.

Zi, Y., Xie, F., Jiang, Z., 2018. A cloud detection method for landsat 8 images based on pcanet. Remote Sens. 10 (6), 877.