Convolutional neural networks for detecting challenging cases in cloud masking using Sentinel-2 imagery

Viktoria Kristollari*[a] and Vassilia Karathanassi[a]

[a]Laboratory of Remote Sensing, School of Rural and Surveying Engineering, National Technical University of Athens, Heroon Polytechniou 9, Zografou, Athens, Greece, 15780

ABSTRACT

Cloud contamination represents a large obstacle for mapping the earth's surface using remotely sensed data. Therefore, cloudy pixels should be identified and eliminated before any further data processing can be achieved. Although several threshold, multi-temporal and machine learning applications have been developed to tackle this issue, it still remains a challenge. The main challenges are imposed by the difficulty to detect thin clouds and to separate bright clouds from bright non-cloud objects. Convolutional neural networks (CNNs) have proven to be one of the most promising methods for image classification tasks and their use is rapidly increasing in remote sensing problems. CNNs present interesting properties for image processing since they directly exploit not only the spectral information but also the spatial covariance of the data. In this work, we study the applicability of CNNs in cloud detection of Sentinel-2 imagery, a complex remote sensing problem with crucial spatial context. A patch-to-pixel CNN architecture consisting of three convolutional layers and two fully connected layers is trained on a recently available manually created public dataset. The results were evaluated both qualitatively and quantitatively through comparison with ground truth cloud masks and state-of-the art pixel-based algorithms (Fmask, Sen2Cor). It was shown that the proposed architecture even though simpler than the deep learning architectures proposed by recent literature, performs very favorably, especially in the challenging cases. Besides the evaluation of the results, feature maps where observed as an initial effort to extract the weights of the useful kernels for cloud masking applications.

Keywords: Convolutional neural networks, Cloud masking, Sentinel satellite imagery, Thin cloud detection, Bright surfaces detection, Feature maps

1. INTRODUCTION

Cloud interference needs to be eliminated for the optimized processing of the acquired optical satellite images. Most of current cloud detection methods extract the clouds from the imagery through ruled based classification which applies a set of thresholds (both static and dynamic) of reflectance and brightness temperature.[1–3] Most widespread threshold methods are ACCA (Automatic Cloud Cover Assessment)[4] and Fmask (Function of mask)[5,6] which was originally designed for Landsat imagery. A threshold based method is also used for the development of the Sentinel-2 cloud masks provided by the level 2A product.[7] Multi-temporal methods based on the idea that abrupt changes in image time series are mainly caused by the presence of clouds have also been extensively implemented.[8–10] MAJA which was designed for Sentinel-2 images is among the most well-known in this category.[11]

The challenging issues in cloud masking include detection of optically thin clouds and separation of bright clouds from non-cloud bright objects (e.g. snow, buildings, desert, coastal sand). Threshold based, multi-temporal and conventional machine learning algorithms struggle to mitigate the above issues. For the detection of high level thin clouds the main approach is the use of thermal bands or the use of the cirrus band (1,375 nm) whenever brightness temperature is unavailable.[5,6] Low level thin clouds are even harder to be detected since their spectral signature is highly similar to the underlying surface. Some indicative studies that report the difficulty in correctly classifying this cloud category were conducted by Zhu and Helmer[12] and Mateo-García et al.[10] who proposed multitemporal methods for Landsat, by Oishi et al.[13] and Zhuge et al.[2] who proposed

*vkristoll@central.ntua.gr

8th International Conference on Remote Sensing and Geoinformation of the Environment, edited by K. Themistocleous, et al. Proc. of SPIE Vol. 11524, 115241N · © The Authors. Published under a Creative Commons Attribution CC-BY 3.0 License · doi: 10.1117/12.2571111

threshold based methods for Landsat and MODIS respectively, and by Gómez-Chova et al.,[14] who tested several conventional machine learning methods for Proba-V.

For the separation of clouds from bright surfaces, operators that extract texture are commonly applied as a pre-processing step, while operators that extract morphology and geometry as a post-processing step. In case of snow, the NDSI index is commonly calculated.[5,6] However, the results are usually in need of improvement as shown from several current research studies implemented in Landsat,[10,12,13] Gaofen-1,[15] Proba-V[16] and MODIS[9] satellites that reported misclassification of bright built-up areas, soils, water bodies (ocean, lake) and snow. Use of a methodology designed for Sentinel-2 that uses parallax has proven most successful till now.[17] Satisfactory results were also produced by an artificial neural network architecture (ANN) that managed to separate sunglint and noise in Sentinel-2 ocean images.[18]

In recent years, convolutional deep learning approaches that use patch-to-pixel or encoder-decoder segmentation architectures have proven successful by taking advantage of the increasing computational power and their inherent ability to perceive spatial information. Convolutional deep learning methods in the majority of studies have produced better and more effortless results than threshold based, multi-temporal and conventional machine learning algorithms. Concerning the detection of thin clouds, Chai et al.[19] proposed an adaptation of Segnet and produced better results compared to CFmask for Landsat images, while Zhaoxiang et al.[20] showed higher accuracy compared to adaboost and random forest by applying a method based on UNET. Successful results were also shown for Quickbird imagery by Yuan et al.[21] who used an encoder-decoder architecture and by Xie et al.,[22] and Shi et al.[23] who used convolutional neural network (CNN) patch-to-pixel architectures. Concerning separation of clouds from bright surfaces without including snow category, Segal et al.[24] proposed a CNN multi-modal patch-to-pixel method for WV-2 and Sentinel-2 imagery and did not observe misclassifications of wave-breaks. Incorrect bright object classification was also not observed by the studies of Zhaoxiang et al.[20] and Li et al.[25] who used encoder-decoder architectures for Landsat and Gaofen-1 imagery respectively. As for the snow category, even convolutional deep learning approaches present difficulties in its separation from clouds.[19,25–27]

From the above, it is clear that convolutional deep learning approaches generally perform better than other approaches in the detection of challenging cases in cloud masking applications. It should be though highlighted that a crucial factor for achieving satisfactory performance is the high accuracy of the ground truth cloud masks. The main technique for creating such masks is visual observation which is time-consuming. Fortunately, Baetens et al.[28] recently created and made publicly available the first public dataset of Sentinel-2 cloud masks which are reported to have 98% accuracy. Thus, this dataset gives the opportunity to perform robust evaluation for Sentinel-2 cloud masking methods. Motivated by this fact, this article proposes a batch-to-pixel CNN architecture for mitigating thin cloud omission and bright non-cloud object commission which pose the main issues in cloud masking applications. For the purpose of the study, different hyperparameters are examined and feature maps are observed. The results are compared qualitatively and quantitatively with ground truth cloud masks and cloud masks produced by state-of-the-art algorithms.

2. PROPOSED METHOD

2.1 Data Description

The study was performed by using in total 37 images collected by Sentinel-2 satellite. The processing level of the images is 1C which denotes that they are not atmospherically corrected. These images where selected on the basis that their respective ground truth cloud masks compose the recently publicly available dataset created by Baetens et al.[28] This dataset is the only publicly available source of Sentinel-2 ground truth cloud masks. Its creation was based on random forest implementation and its accuracy is reported to be 98%. The images depict several areas around the world with high land cover and cloud variability. The images were collected in: Europe (19), North America (three), South America (four), Africa (10) and Australia (one). The dates of the collection cover all seasons of the year: eight winter images (December, January, February), eight spring images (March, April, May), 12 summer images (June, July, August) and nine fall images (September, October, November). The collection time varies between seven a.m. and six p.m UTC.

Sentinel-2 images contain 13 bands with spatial resolution 60 m (three bands), 10 m (four bands) and 20 m (six bands). The wavelengths of the 3 spatial resolutions of the Sentinel-2 instruments are shown in Table 1. Before analysis, these images were processed. The bands with spatial resolution 10 and 20 m were resampled to 60 m, with x-size (columns):1,830 pixels and y-size (rows):1,830 pixels. Then, zero padding (size=eight pixels) was added around the images so that the size of the cloud masks produced by the CNN is the same as the Sentinel-2 images, since the input patch x-size and y-size was 16x16. The training set consisted of 16 images and the test set of 21. Good representation of land cover and cloud variability was the main factor that was taken into account when selecting the images of the training set. Focus was also put on the inclusion of adequate samples of thin clouds and bright non-cloud objects.

Table 1. Wavelengths of the three spatial resolutions of Sentinel-2

Spatial resolution (m)	Band number	S2A Central wavelength (nm)	S2B Central wavelength (nm)
10	2	496.6	492.1
	3	560	559
	4	664.5	665
	8	835.1	833
20	5	703.9	703.8
	6	740.2	739.1
	7	782.5	779.7
	8A	864.8	864
	11	1613.7	1610.4
	12	2202.4	2185.7
60	1	443.9	442.3
	9	945	943.2
	10	1373.5	1376.9

2.2 CNN Architecture

The patch-to-pixel CNN architecture proposed in this study is composed of three convolutional layers and three pooling layers. Each convolutional layer is followed by a pooling layer that retains the maximum value of a window with size 2x2. The patch input size of the CNN is 16x16 and the output predicts the central pixel of the patch. A kernel of size 3x3 is applied for all three convolutional layers. It was decided to use zero padding before applying convolution, thus the size of the output of the operation is the same as the size of the input. The convolution operation is depicted in Equation (1). The CNN architecture is followed by a flattening layer, two fully connected layers each of which is composed by 50 neurons and an output layer. This architecture was investigated by implementing three different versions. In the first version (Fig. 1) the Leaky Rectified Linear Unit (Leaky ReLU) activation function was applied after all three convolutional layers. The difference of this function from ReLU (Equation (2))[29] is the use of a small slope for negative values instead of zero. In the second version the ReLU activation function was applied and in the third batch normalization (BN)[30] which normalizes input layers was combined with Leaky ReLU (BN was applied before Leaky ReLU). For all three versions, the ReLU function was used in the two fully connected layers and the sigmoid function (Equation (3))[31] in the output layer. In addition, dropout method[32] which ignores neurons at random and prevents overfitting (value=0.3) was applied in the two fully connected layers.

$$G[i,j] = h * F = \sum_{u=-k}^{k} \sum_{v=-k}^{k} h[u,v] F[i-u, j-v]\,, \tag{1}$$

where h is the image, F is the filter. u, v are row and collumn coordinates of the image and i, j are row and collumn coordinates of the filter.

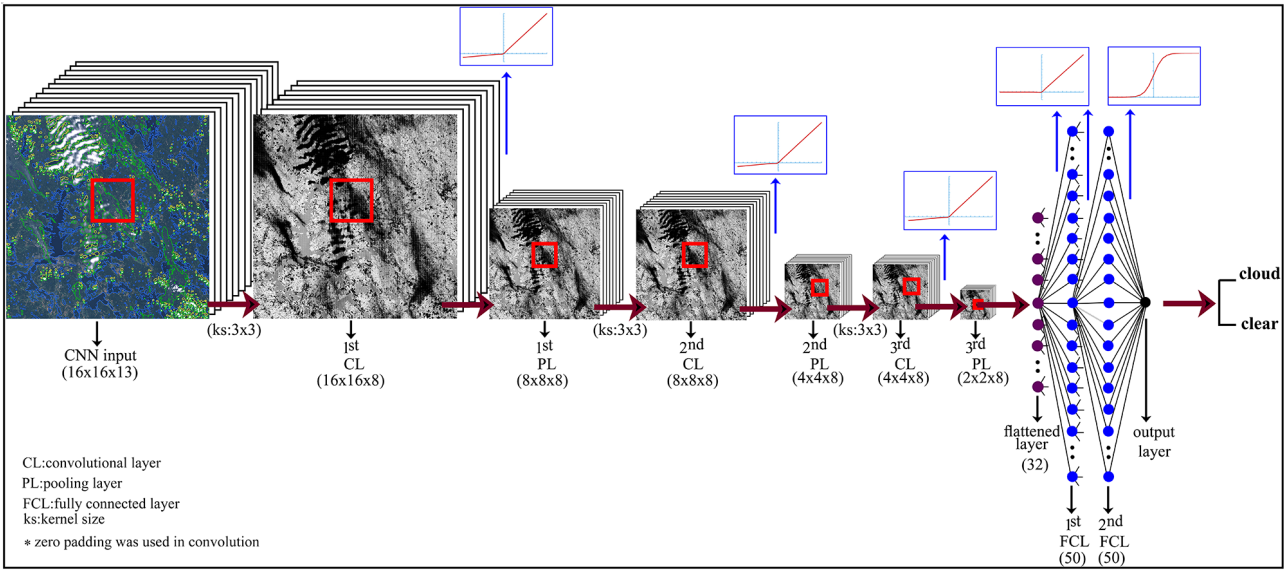

Figure 1. Architecture of the first version of the proposed CNN (Use of Leaky ReLU)

$$\phi(x) = \max(0, x) \tag{2}$$

$$\phi(x) = \frac{1}{1 + e^{-x}} \tag{3}$$

where x is either the output produced by the convolution operation (convolutional layer) or the sum of the product of the weights connecting two layers with the input (fully connected layer).

2.3 Training and Inference

The training was performed on an NVIDIA Graphic Processing Unit (GPU) (NVIDIA 1070 Ti) using Keras library[33] with Tensorflow[34] as backend. Each of the three CNN models was trained for 30 epochs with 10,000 train steps. Training time was similar for the model that used Leaky ReLU and the model that used ReLU and it lasted approximately nine hours (the training time for the model that used ReLU was slightly faster). The training time for the model that used BN was 12 hours. Inference time was approximately two minutes for all models. A generator function was designed for the training with the purpose to feed the CNN with batches of training data. Every time the generator was called, it selected randomly one of the 16 images of the training set and then it selected all the pixels of a random line as central pixels of a patch of size (16x16x13) where 13 is the number of the Sentinel-2 bands. A similar generator was designed for the 21 images of the test set in order to compute accuracy and loss values for every epoch. During training, the weights were updated by applying Adaptive moment estimation (Adam)[35] with Equation (4) as the loss function. Adam stores an exponentially decaying average of past squared gradients $\boldsymbol{v}_t$ (Equation (6)) and an exponentially decaying average of past gradients $\boldsymbol{m}_t$(Equation (5)). The gradients $\boldsymbol{g}_t$ denote the vector of partial derivatives of the loss function at timestep t. The zero bias of $\boldsymbol{m}_t$ and $\boldsymbol{v}_t$ is counteracted by computing bias-corrected first and second moment estimates (Equations (7, 8)). These are used to update the weights (Equation (9)).

$$H_p(q) = -\frac{1}{N}\sum_{i=1}^{N} y_i \cdot log(p(y_i)) \cdot log(1 - p(y_i)) \tag{4}$$

where y is the label and $p(y)$ is the probability of the central pixel of the input patch being classified as cloud.

$$\boldsymbol{m}_t = \beta_1 \boldsymbol{m}_{t-1} + (1-\beta_1)\boldsymbol{g}_t \tag{5}$$

$$\boldsymbol{v}_t = \beta_2 \boldsymbol{v}_{t-1} + (1-\beta_2)\boldsymbol{g}_t^2 \tag{6}$$

$$\hat{\boldsymbol{m}}_t = \frac{\boldsymbol{m}_t}{1-\beta_1^t} \tag{7}$$

$$\hat{\boldsymbol{v}}_t = \frac{\boldsymbol{v}_t}{1-\beta_2^t} \tag{8}$$

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \frac{\eta}{\sqrt{\hat{\boldsymbol{v}}_{\boldsymbol{t}}} + \epsilon}\hat{\boldsymbol{m}}_t \tag{9}$$

where β_1 and β_2 are exponential decay rates for the moment estimates and η is the learning rate.

3. RESULTS

3.1 Training

The values of accuracy and loss function for the training and the test sets during 30 epochs are shown in Fig. 2 for the three CNN models. In addition, the average accuracy and loss values are shown in Table 2. It can be observed that the CNN models trained by use of the Leaky ReLU and ReLU activation functions demonstrated high accuracy ($\sim$ 96%) and low loss values (<0.12) for both the training and the test. In contrast, the model that combined BN with Leaky ReLU performed by far less favorably since it showed high instability. In more detail, as it can be seen in Fig. 2, accuracy and loss values of the training set showed large differences since the former ranged between $\sim$ 0.75 and $\sim$ 0.95, and the latter between $\sim$ 0 and $\sim$ 2.5. Also, the lower performance of this model can be seen by the large difference of the average values of accuracy and loss function of the training set compared to the test set ($\sim$ 90%, $\sim$ 96% and $\sim$ 0.33, $\sim$ 0.10). By observing the plots of Fig. 2, it was decided to produce cloud masks only by use of the Leaky ReLU model of the last epoch since the accuracy of the test set for this epoch was slightly better than ReLU (Table 3).

Table 2. Average values of accuracy and loss (30 epochs) for the training and test sets

CNN model	Accuracy		Loss	
	Training set	Test set	Training set	Test set
Leaky ReLU	0.9551	0.9612	0.1166	0.1442
ReLU	0.9539	0.9577	0.1192	0.0944
BN + Leaky ReLU	0.9614	0.8961	0.0993	0.3279

Table 3. Last epoch values of accuracy and loss for the training and test sets

CNN model	Accuracy		Loss	
	Training set	Test set	Training set	Test set
Leaky ReLU	0.9638	0.9621	0.0939	0.0032
ReLU	0.9633	0.9549	0.0941	0.0148
BN + Leaky ReLU	0.9682	0.8884	0.082	0.0667

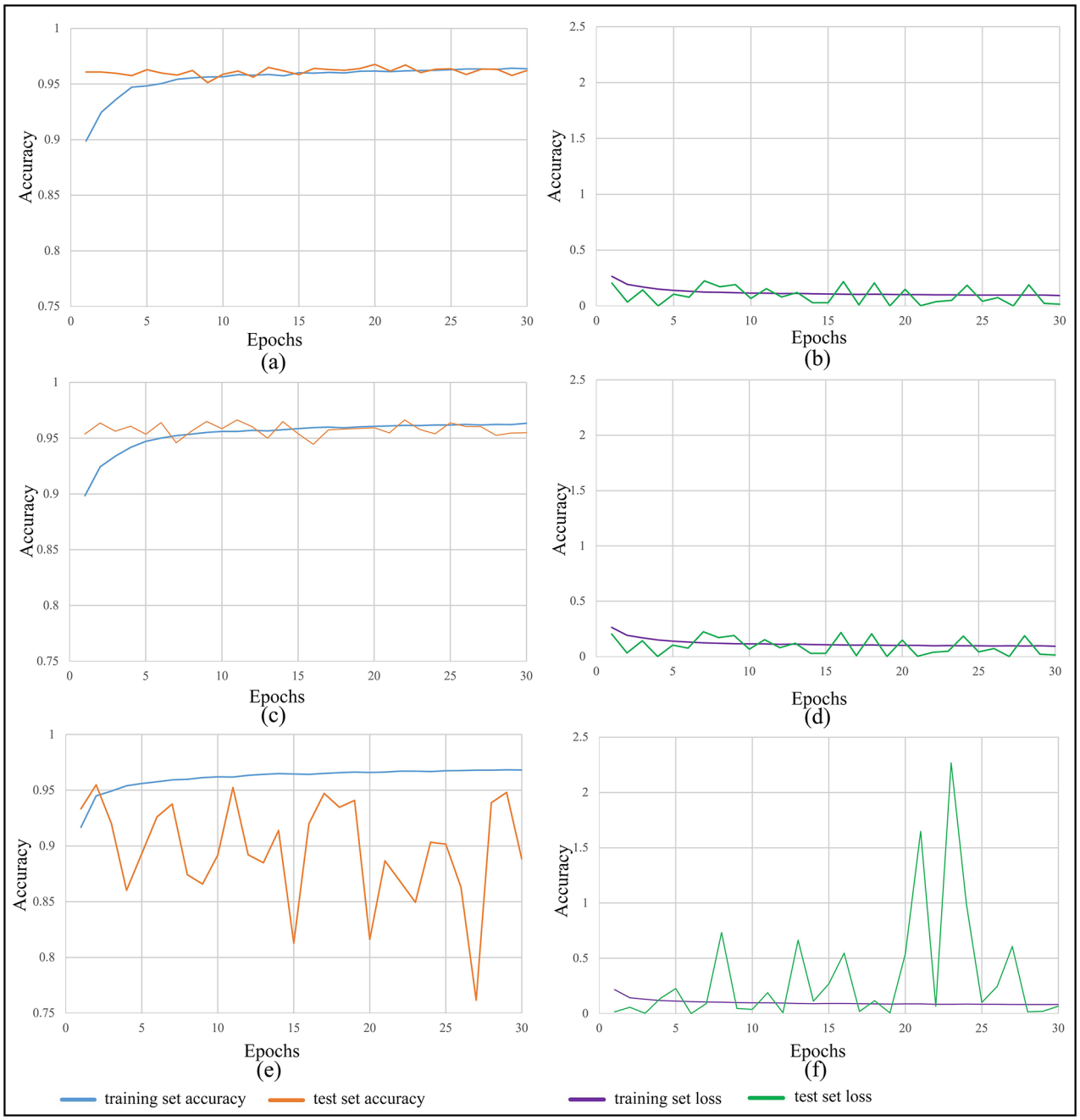

Figure 2. (a,b):Accuracy/Loss of model trained with Leaky ReLU, (c,d):Accuracy/Loss of model trained with ReLU, (e,f):Accuracy/Loss of model trained with BN and Leaky ReLu

3.2 Sentinel-2 Cloud Masks

Evaluation metrics were computed for the cloud masks produced by the model trained with the Leaky ReLU and the respective cloud masks produced by Sen2Cor and Fmask. The metrics were calculated by considering as ground truth masks those of the dataset produced by Baetens et al.[28] The average values are presented in Table 4 and the values for each of the 37 images (16 training images, 21 test images) are presented in Fig. 3. The metrics that were computed were accuracy (Equation (10)), recall (producer's accuracy)(Equation (11)),

precision (user's accuracy)(Equation (12)) and fscore (Equation (13)). Recall corresponds to omission error (1-omission error) while precision corresponds to commission error (1-commission error).

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FN} + \text{FP} + \text{TN}} \tag{10}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{11}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{12}$$

$$\text{Fscore} = 2 \text{ x } \frac{\text{precision x recall}}{\text{precision} + \text{recall}} \tag{13}$$

where TP: True positive, TN: True negative, FP: False positive, FN: False negative.

Table 4. Evaluation metrics of Sentinel-2 cloud masks

Method	Accuracy	Recall	Precision	Fscore
Sen2Cor	0.9170	0.9215	0.9713	0.9412
Fmask	0.9193	0.9115	0.9856	0.9424
CNN (training set)	0.9742	0.9762	0.9847	0.9804
CNN (test set)	0.9751	0.9800	0.9815	0.9805

For the CNN model, the evaluation metrics were calculated separately for the training and test sets. It was observed that the CNN showed exceptional performance both in the training set and the test set with all evaluation metrics having values $\sim$ 98%. Concerning the state-of-the-art algorithms, the accuracy and recall values of Sen2Cor and Fmask were $\sim$ 92%, the precision values were $\sim$ 98% and the fscore values were $\sim$ 94%. Thus, these two state-of-the-art algorithms performed similarly and by far less favorably than the CNN model. The same conclusion can be reached by observing Fig. 3 and the box plots of Fig. 4. Box plots are diagrams that show the variance of the data. Each box plot is formed by two boxes. The lower side of the lower box denotes the first quartile and the upper side the second quartile. The upper side of the upper box denotes the third quartile. The vertical lines in the middle of the boxes show the distance of the maximum or minimum value compared to the second quartile. In the box plots of this study it can be seen that for the CNN the values of all evaluation metrics are much closer to the mean value compared to Sen2Cor and Fmask.

3.3 Challenging cases

Fig. 5 and Fig. 6 present the cloud masks produced for indicative challenging cases by Sen2Cor, Fmask and the CNN for the training set and the test set respectively. These figures show the RGB natural composite with delineation of the ground truth categories by Baetens et al.[28] as well as correctly predicted pixels (for categories of cloud (TP) and clear (TN)) along with omission (FN) and commission error (FP). The evaluation metrics for these particular cases are stated in Tables 5, 6 for the training set and the test set respectively.

Concerning the challenging cases of the training set, Fig. 5(a1, a2) depict cases with optically thin clouds where high percentage is characterized by very high transparency. It is obvious that the omission error of the CNN is very small for Fig. 5(a1)(<3%)(Fig. 5(d1) in contrast to Sen2Cor ($\sim$ 13%)(Fig. 5(b1)) and Fmask ($\sim$ 11%)(Fig. 5(c1)). Similarly, the omission error for the CNN cloud mask of Fig. 5(a2) is much smaller ($\sim$ 8%) (Fig. 5(d2)) than the respective cloud masks of Sen2Cor ($\sim$ 33%)(Fig. 5(b2)) and Fmask ($\sim$ 40%) (Fig. 5(c2)). Fig. 5(a3, a4) depict cases of non-cloud bright objects. From the produced cloud masks it can be observed that the snow area of Fig. 5(a3) is correctly classified by the CNN (Fig. 5(d3)) while the other two methods incorrectly detect this snow area as cloud (Fig. 5(b3, c3)). It can also be seen that the CNN shows much smaller omission error than the other algorithms. As for Fig. 5(a4), it can be observed that the bright non-cloud area

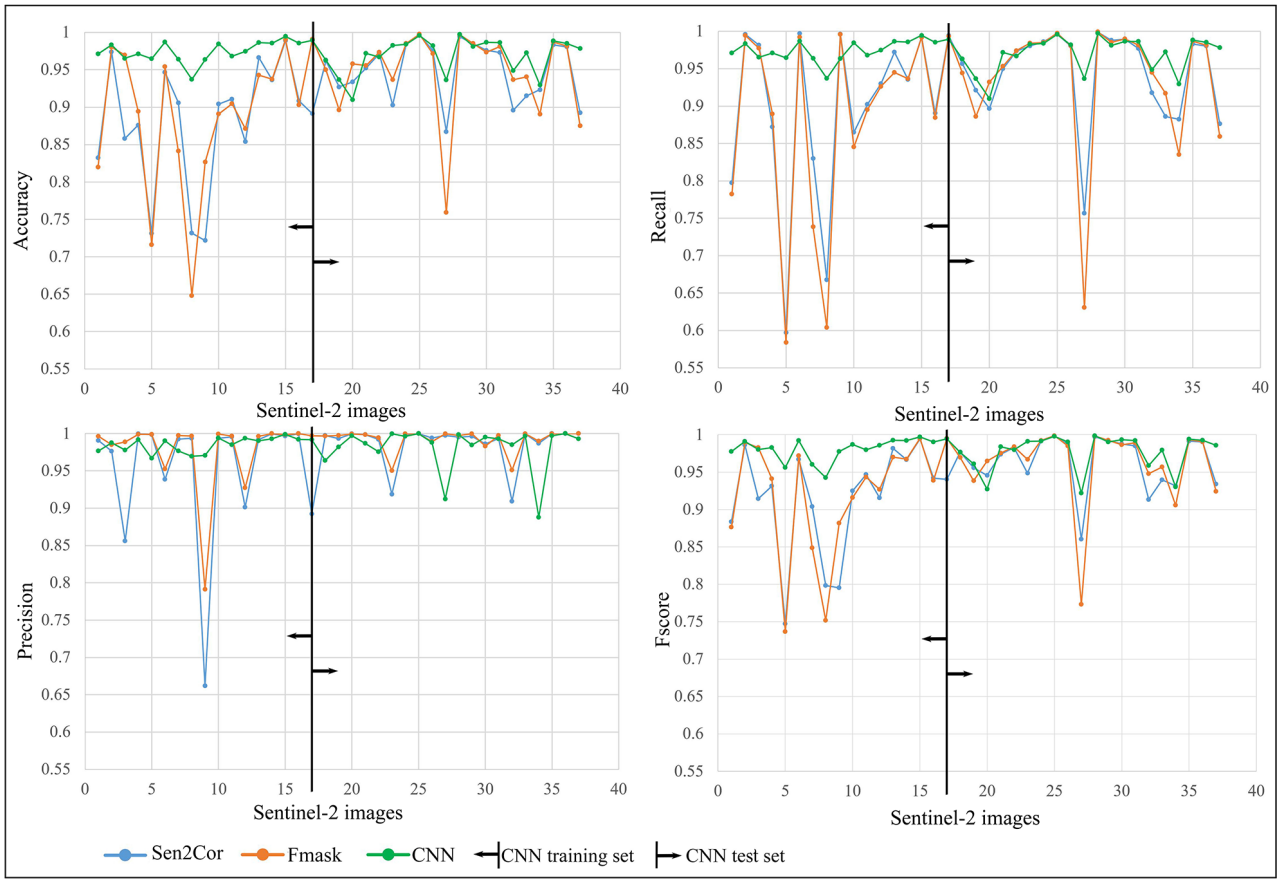

Figure 3. Evaluation metrics of the Sentinel-2 images

is successfully classified by the CNN, (Fig. 5(d4)) while Sen2Cor (Fig. 5(b4)) and Fmask (Fig. 5(c4)) fail to correctly categorize it.

Similar conclusions can be reached for the challenging cases of the test set (Fig. 6). Fig. 6(a1) presents a case of semi-transparent clouds where the CNN cloud mask (Fig. 6(d1)) shows very low omission error (∼2%) in contrast to Sen2Cor (Fig. 6(b1)) and Fmask (Fig. 6(c1)) which produce larger omission errors (∼12%, ∼14%). Fig. 6(a2) presents a region with snow mountainous areas and an extended bright urban area. For this image, Sen2Cor (Fig. 6(b2)) and Fmask (Fig. 6(c2)) produce cloud masks that incorrectly classify a large part of the snow area as cloud and also incorrectly classify some bright urban elements (shown in zoom out circle). The respective CNN cloud mask (6(d2)) performs more successfully both in the snow and in the urban area. In Fig. 6(a3) it can be observed that the CNN can detect more cloud areas that have similar spectral signatures with the background (Fig. 6(d3)) in contrast to the other two methods (Fig. 6(b3, c3)). Finally, regarding the bright non-cloud objects of Fig. 6(a4), it can be seen that CNN (Fig. 6(d4)) and Fmask (Fig. 6(c4) perform similarly while Sen2Cor (Fig. 6(b4)) produces a high commission error.

3.4 Feature Maps

Besides training the CNN, this study did an initial effort to investigate the feature maps produced by the convolutional layers, since it would be useful to extract kernels that could be used for the production of features for cloud masking. These kernels could potentially form a database that would enhance performance of feature-based cloud masking methods. Fig. 7 depicts an indicative example of an image of the training set which represents a very difficult case for cloud masking since it contains clouds of very high transparency. From visual observation, it can be assumed that the feature map of Fig. 7(a6) manages to detect more successfully this

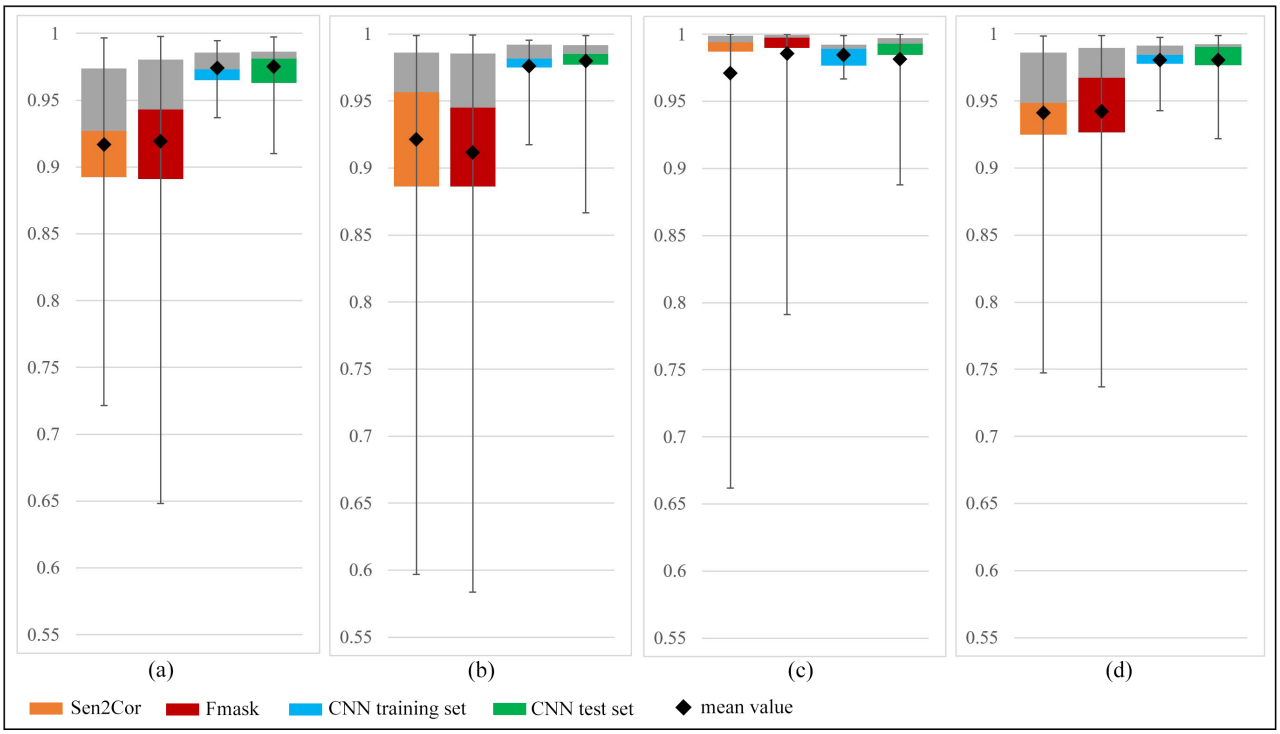

Figure 4. Box plots of the evaluation metrics of the Sentinel-2 images. (a):Accuracy, (b):Recall, (c):Precision, (d):Fscore

Table 5. Evaluation metrics of the challenging cases of the training set

Fig.	Accuracy			Recall			Precision			Fscore		
	S2cor	Fmask	CNN	S2Cor	Fmask	CNN	S2Cor	Fmask	CNN	S2Cor	Fmask	CNN
a1	0.876	0.8943	0.9711	0.872	0.8894	0.9745	0.9998	0.999	0.9917	0.9315	0.941	0.983
a2	0.7317	0.6482	0.937	0.6675	0.6038	0.9173	0.9938	0.9966	0.9697	0.7986	0.752	0.9428
a3	0.8539	0.8714	0.9747	0.93	0.9263	0.9779	0.9014	0.9274	0.9936	0.9155	0.9268	0.9857
a4	0.9466	0.9541	0.987	0.9971	0.9921	0.9939	0.9387	0.9527	0.9905	0.967	0.972	0.9922

type of clouds. Fig. 7 also shows the 13 kernels that were used in the convolution operation that produced the above mentioned feature map. As already stated, a database composed by kernels of such kind could give the opportunity to easily recreate the feature maps without the need to have any prior information about the CNN. The images that these kernels would be applied should of course depict similar spectral range and potentially similar land cover to increase effectiveness.

4. CONCLUSION

This study proposed a CNN model that succesfully detects semi-transparent clouds and separates bright clouds from bright non-cloud objects. The proposed method is applied on the first publicly available dataset of Sentinel-2 ground truth cloud masks which provides the opportunity for a robust and objective evaluation. Different versions of the proposed CNN architecture were investigated with the version using the Leaky ReLU activation fuction showing slightly higher accuracy in the test set than the version that used ReLU. The version that used BN produced the less accurate and more unstable results. The Leaky ReLU version was evaluated in the training and test sets quantitatively by calculating four evaluation metrics and qualitatively by visually observing the produced cloud masks. Comparison with cloud masks produced by Sen2Cor and Fmask was performed for both

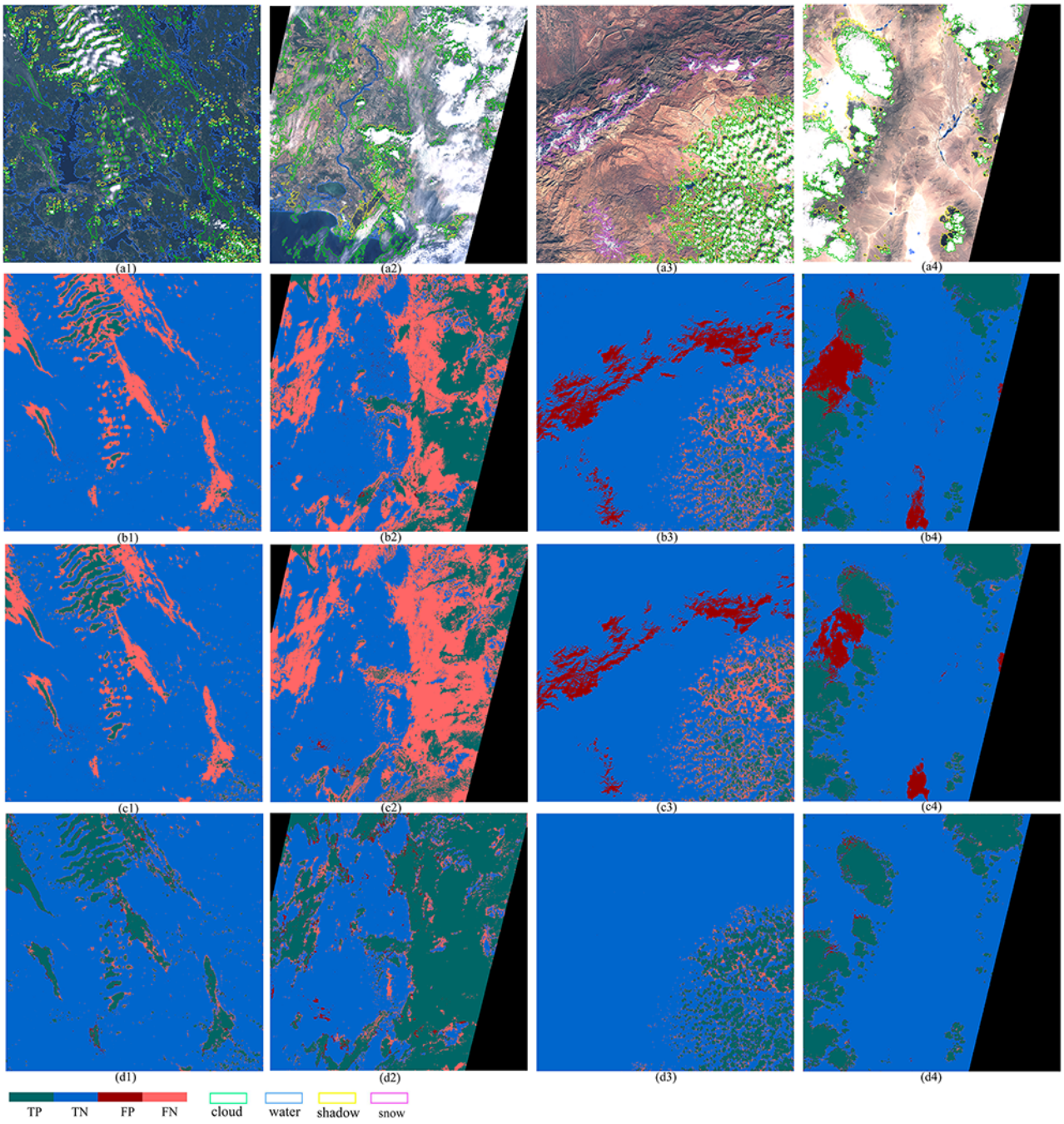

Figure 5. Cloud masks of the challenging cases of the training set. (a1-a4):RGB composite with delineation of categories, (b1-b4):Sen2Cor cloud masks, (c1-c4):Fmask cloud masks, (d1-d4):CNN cloud masks

evaluations.

It was shown that CNN produced exceptional results ($\sim 98\%$) both in the training and the test set compared to the state-of-the-art threshold-based methods which performed by far less favorably. In more detail, the CNN managed to detect even clouds of very high transparency and successfully separated clouds from snow as well as bright urban and desert areas. Thus, the study further reinforces the value of CNNs in applications where

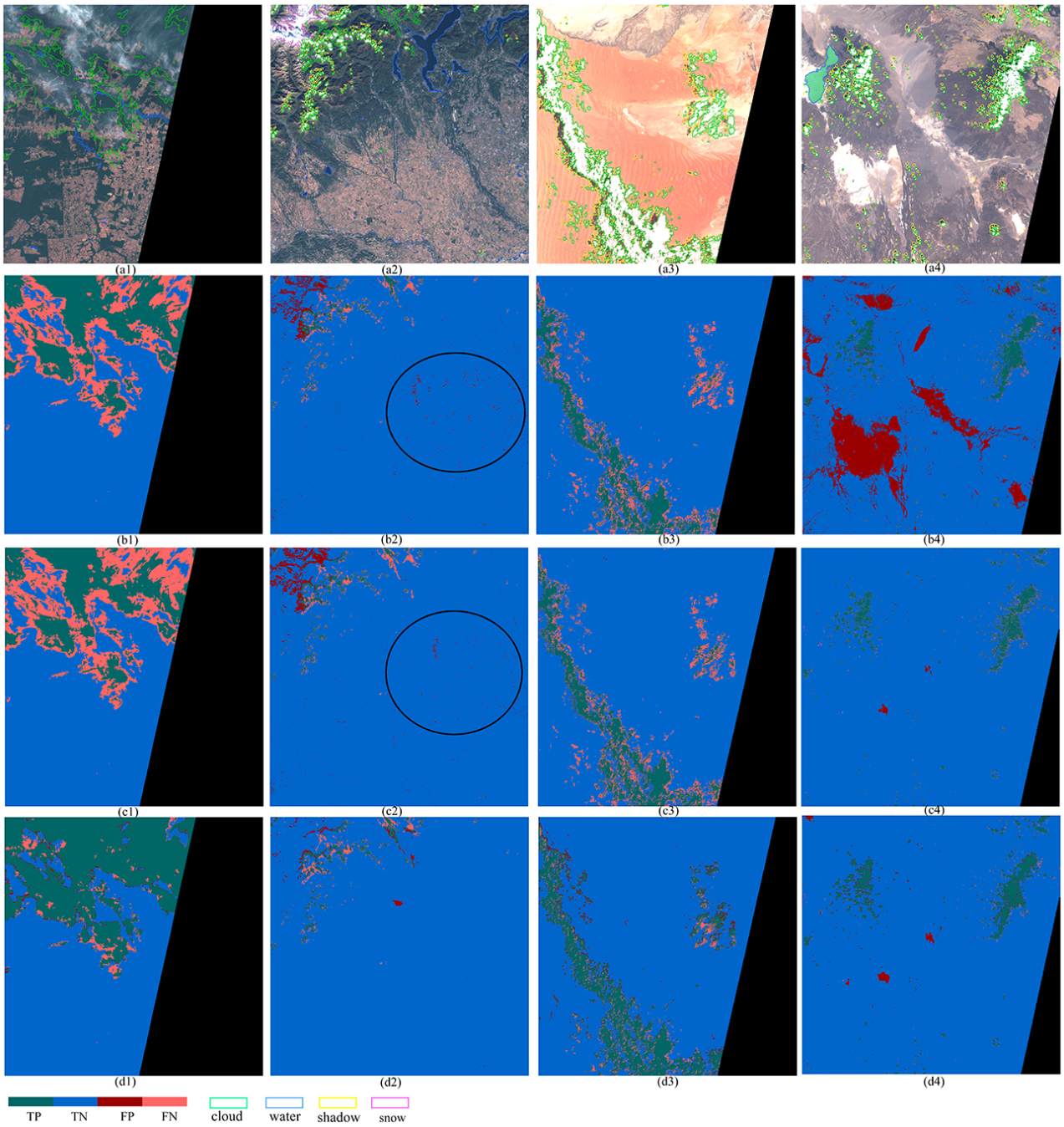

Figure 6. Cloud masks of the challenging cases of the test set.(a1-a4):RGB composite with delineation of categories, (b1-b4):Sen2Cor cloud masks, (c1-c4):Fmask cloud masks, (d1-d4):CNN cloud masks

spatial context is very important, and shows that an architecture that makes use of smaller number of layers and feature maps compared to recent deep learning literature, consequently being simpler and more time-efficient, can produce very satisfactory results in cloud masking.

Besides observing the produced cloud masks, an initial effort was performed to observe the feature maps produced by the convolutional layers aiming to extract the weights of the kernels. In our opinion, a database

Table 6. Evaluation metrics of the challenging cases of the test set

Fig.	Accuracy			Recall			Precision			Fscore		
	S2Cor	Fmask	CNN	S2Cor	Fmask	CNN	S2Cor	Fmask	CNN	S2Cor	Fmask	CNN
a1	0.8926	0.8752	0.9782	0.8764	0.8593	0.9788	1	1	0.9929	0.9341	0.9243	0.9858
a2	0.9759	0.9736	0.9866	0.9894	0.9898	0.9912	0.986	0.9831	0.9951	0.9877	0.9865	0.9932
a3	0.9524	0.9555	0.9719	0.9499	0.9533	0.9813	0.9988	0.9986	0.9869	0.9737	0.9754	0.9841
a4	0.8914	0.9909	0.9893	0.9944	0.9933	0.9971	0.8922	0.9973	0.9918	0.9405	0.9953	0.9944

formed by such kernels would be very useful since it can easily provide crucial features that could be input to several algorithms outside of the context of neural networks. The creation of such a database is intended to be part of our future work. Training and testing the method with more data (given that more ground truth cloud masks will be created) is also intended since the land cover of the Earth shows very large variability.

5. ACKNOWLEDGMENTS

This research was conducted in the framework of the SEO-DWARF project which is supported by H2020 Marie Sklodowska-Curie Actions (http://dx.doi.org/10.13039/100010665). More details about the project can be found at https://cordis.europa.eu/project/id/691071.

REFERENCES

[1] Wilson, M. J. and Oreopoulos, L., "Enhancing a simple MODIS cloud mask algorithm for the landsat data continuity mission," *IEEE Transactions on Geoscience and Remote Sensing* **51**, 723–731 (Jul. 2013).

[2] Zhuge, X.-Y., Zou, X., and Wang, Y., "A fast cloud detection algorithm applicable to monitoring and nowcasting of daytime cloud systems," *IEEE Transactions on Geoscience and Remote Sensing* **55**, 6111–6119 (Nov. 2017).

[3] Jedlovec, G., Haines, S., and LaFontaine, F., "Spatial and Temporal Varying Thresholds for Cloud Detection in GOES Imagery," *IEEE Transactions on Geoscience and Remote Sensing* **46**, 1705–1717 (Jun. 2008).

[4] Irish, R. R., "Landsat 7 automatic cloud cover assessment," in [*Algorithms for Multispectral, Hyperspectral, and Ultraspectral Imagery VI*], **4049**, 348–355, International Society for Optics and Photonics (Aug. 2000).

[5] Zhu, Z. and Woodcock, C. E., "Object-based cloud and cloud shadow detection in landsat imagery," *Remote sensing of environment* **118**, 83–94 (Mar. 2012).

[6] Zhu, Z., Wang, S., and Woodcock, C. E., "Improvement and expansion of the fmask algorithm: Cloud, cloud shadow, and snow detection for landsats 4–7, 8, and sentinel 2 images," *Remote Sensing of Environment* **159**, 269–277 (Mar. 2015).

[7] Richter, R., Louis, J., and Müller-Wilm, U., "Sentinel-2 msilevel 2a products algorithm theoretical basis document," *European Space Agency,(Special Publication) ESA SP* **49**(0), 1–72 (2012).

[8] Hagolle, O., Huc, M., Pascual, D. V., and Dedieu, G., "A multi-temporal method for cloud detection, applied to formosat-2, vens, landsat and sentinel-2 images," *Remote Sensing of Environment* **114**, 1747–1755 (Aug. 2010).

[9] Karvonen, J., "Cloud masking of modis imagery based on multitemporal image analysis," *International Journal of Remote Sensing* **35**, 8008–8024 (Dec. 2014).

[10] Mateo-García, G., Gómez-Chova, L., Amorós-López, J., Muñoz-Marí, J., and Camps-Valls, G., "Multitemporal Cloud Masking in the Google Earth Engine," *Remote Sensing* **10**, 1079 (Jul. 2018).

[11] Hagolle, O., Huc, M., Desjardins, C., Auer, S., and Richter, R., "Maja atbd algorithm theoretical basis document," *tech. rep., CNES+ CESBIO and DLR* (2017).

[12] Zhu, X. and Helmer, E. H., "An automatic method for screening clouds and cloud shadows in optical satellite image time series in cloudy regions," *Remote sensing of environment* **214**, 135–153 (Sep. 2018).

[13] Oishi, Y., Ishida, H., and Nakamura, R., "A new landsat 8 cloud discrimination algorithm using thresholding tests," *International journal of remote sensing* **39**, 9113–9133 (Dec. 2018).

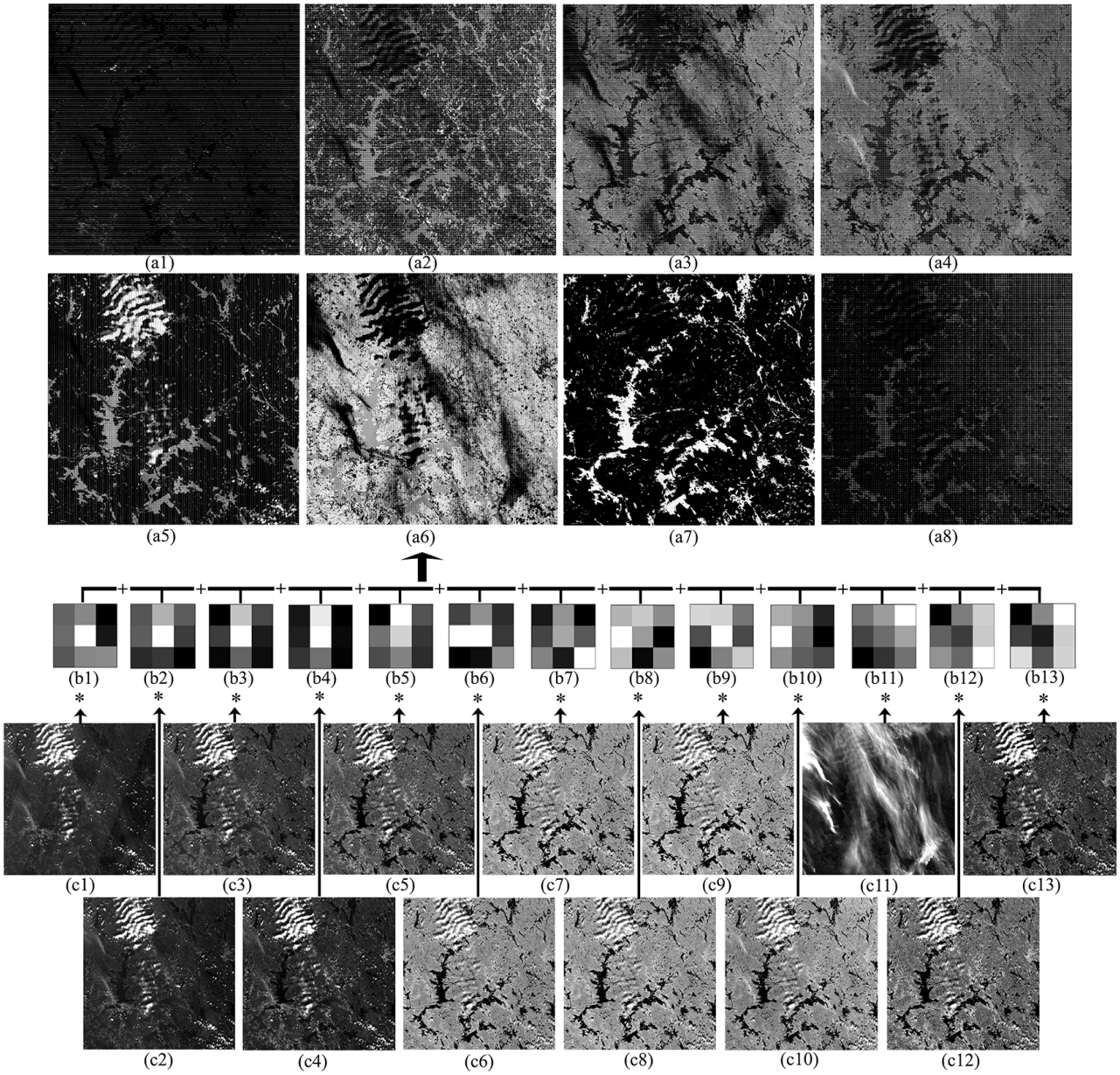

Figure 7. (a1-a8):Feature maps of the first convolutional layer for an indicative example, (b1-b13):Kernels used in the convolution operation that produced a6, (c1-c13):Sentinel-2 bands

[14] Gómez-Chova, L., Mateo-García, G., Muñoz-Marí, J., and Camps-Valls, G., "Cloud detection machine learning algorithms for proba-v," in [*2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*], 2251–2254, IEEE (Jul. 2017).

[15] Li, Z., Shen, H., Li, H., Xia, G., Gamba, P., and Zhang, L., "Multi-feature combined cloud and cloud shadow detection in gaofen-1 wide field of view imagery," *Remote sensing of environment* **191**, 342–358 (March. 2017).

[16] Iannone, R., Niro, F., Goryl, P., Dransfeld, S., Hoersch, B., Stelzer, K., Kirches, G., Paperin, M., Brockmann, C., Gómez-Chova, L., et al., "Proba-v cloud detection round robin: Validation results and recom-

mendations," in [*2017 9th International Workshop on the Analysis of Multitemporal Remote Sensing Images (MultiTemp)*], 1–8, IEEE (Jun. 2017).

[17] Frantz, D., Haß, E., Uhl, A., Stoffels, J., and Hill, J., "Improvement of the fmask algorithm for sentinel-2 images: Separating clouds from bright surfaces based on parallax effects," *Remote sensing of environment* **215**, 471–481 (Mar. 2018).

[18] Kristollari, V. and Karathanassi, V., "Artificial neural networks for cloud masking of sentinel-2 ocean images with noise and sunglint," *International Journal of Remote Sensing* **41**, 4102–4135 (Jan. 2020).

[19] Chai, D., Newsam, S., Zhang, H. K., Qiu, Y., and Huang, J., "Cloud and cloud shadow detection in landsat imagery based on deep convolutional neural networks," *Remote sensing of environment* **225**, 307–316 (Sep. 2019).

[20] Zhaoxiang, Z., Iwasaki, A., Guodong, X., and Jianing, S., "Small satellite cloud detection based on deep learning and image compression," 1–12 (Feb. 2018).

[21] Yuan, K., Meng, G., Cheng, D., Bai, J., Xiang, S., and Pan, C., "Efficient cloud detection in remote sensing images using edge-aware segmentation network and easy-to-hard training strategy," in [*2017 IEEE International Conference on Image Processing (ICIP)*], 61–65, IEEE (Sep. 2017).

[22] Xie, F., Shi, M., Shi, Z., Yin, J., and Zhao, D., "Multilevel cloud detection in remote sensing images based on deep learning," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* **10**, 3631–3640 (Aug. 2017).

[23] Shi, M., Xie, F., Zi, Y., and Yin, J., "Cloud detection of remote sensing images by deep learning," in [*2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*], 701–704, IEEE (Jul. 2016).

[24] Segal-Rozenhaimer, M., Li, A., Das, K., and Chirayath, V., "Cloud detection algorithm for multi-modal satellite imagery using convolutional neural-networks (cnn)," *Remote Sensing of Environment* **237**, 111446 (Feb. 2020).

[25] Li, Z., Shen, H., Wei, Y., Cheng, Q., and Yuan, Q., "Cloud detection by fusing multi-scale convolutional features," *Proceedings of the ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Science, Beijing, China* , 7–10 (Apr. 2018).

[26] Lu, J., Wang, Y., Zhu, Y., Ji, X., Xing, T., Li, W., and Zomaya, A. Y., "P_segnet and np_segnet: New neural network architectures for cloud recognition of remote sensing images," *IEEE Access* **7**, 87323–87333 (Jul. 2019).

[27] Xia, M., Liu, W., Shi, B., Weng, L., and Liu, J., "Cloud/snow recognition for multispectral satellite imagery based on a multidimensional deep residual network," *International journal of remote sensing* **40**, 156–170 (Aug. 2019).

[28] Baetens, L., Desjardins, C., and Hagolle, O., "Validation of copernicus sentinel-2 cloud masks obtained from maja, sen2cor, and fmask processors using reference cloud masks generated with a supervised active learning procedure," *Remote Sensing* **11**, 433 (Feb 2019).

[29] Agarap, A. F., "Deep learning using rectified linear units (relu)," *arXiv preprint arXiv:1803.08375* (2018).

[30] Ioffe, S. and Szegedy, C., "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167* (2015).

[31] Nwankpa, C., Ijomah, W., Gachagan, A., and Marshall, S., "Activation functions: Comparison of trends in practice and research for deep learning. arxiv 2018," *arXiv preprint arXiv:1811.03378* (2018).

[32] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R., "Dropout: a simple way to prevent neural networks from overfitting," *The journal of machine learning research* **15**, 1929–1958 (Jun. 2014).

[33] F, C., "Keras." https://github.com/fchollet/keras (2015).

[34] Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., et al., "Tensorflow: A system for large-scale machine learning," in [*12th Symposium on Operating Systems Design and Implementation*], 265–283 (2016).

[35] Kingma, D. P. and Ba, J., "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980* (2014).