

CONVOLUTIONAL NEURAL NETWORKS FOR MULTISPECTRAL IMAGE CLOUD MASKING

Gonzalo Mateo-García, Luis Gómez-Chova, Gustau Camps-Valls

Image Processing Laboratory (IPL), University of Valencia, Spain

ABSTRACT

Convolutional neural networks (CNN) have proven to be *state of the art* methods for many image classification tasks and their use is rapidly increasing in remote sensing problems. One of their major strengths is that, when enough data is available, CNN perform an *end-to-end* learning without the need of custom feature extraction methods. In this work, we study the use of different CNN architectures for cloud masking of Proba-V multispectral images. We compare such methods with the more classical machine learning approach based on feature extraction plus supervised classification. Experimental results suggest that CNN are a promising alternative for solving cloud masking problems.

Index Terms— Convolutional neural networks, deep learning, cloud masking, cloud detection, Proba-V

1. INTRODUCTION

In the last years, convolutional neural networks (CNN) have become one of the most promising methods for both general image classification tasks [1,2] and also remote sensing image classification [3–5]. Beyond the high classification accuracy shown in many problems, CNN present interesting properties for remote sensing image processing since they directly learn from the available data the most relevant spatial features for the given problem, i.e. a previous custom feature extraction step is not required [6]. In this paper, we analyze the applicability of different CNN architectures in a complex remote sensing problem in which the spatial context is of paramount importance: cloud masking of Proba-V multispectral imagery.

Images acquired by the Proba-V instrument [7], which works in the visible and infrared (VIS-IR) ranges of the electromagnetic spectrum, may be affected by the presence of clouds. Cloud masking can be tackled as a two-class classification problem; and the simplest approach to cloud detection in a scene is the use of a set of static thresholds (e.g.

over reflectance or temperature) applied to every pixel in the image, which provides a cloud flag (binary classification). However, Proba-V instrument presents a limited number of spectral bands (Blue, Red, NIR and SWIR) which makes cloud detection particularly challenging since it does not present thermal channels or a dedicated cirrus band. Current Proba-V cloud detection uses multiple thresholds applied to the blue and the SWIR spectral bands [8], but the definition of global thresholds is practically impossible. Hence, for next Proba-V reprocessing [9], monthly composites of cloud-free reflectance in the blue band are used to define dynamic thresholds depending on the land cover type. In this context, given the reduced amount of spectral information, spatial information seems crucial to increase the performance of classification methods and the cloud detection accuracy.

2. METHODOLOGY

CNN are a special type of neural networks that present a series of convolutional layers especially designed to cope with inputs in the form of multidimensional arrays (image patches) [10]. The CNN architecture used in this work is based on [11] and consists of 2 blocks of 2 convolutional layers followed by a max-pooling layer. Each convolutional layer is formed by convolution, batch normalization [12], and a rectified linear unit (ReLU). At the top a fully connected (FC) block with 256 hidden units is included, whose outputs are used to predict the output with a sigmoid activation function.

The network is trained to minimize the binary cross-entropy between predictions $h(\mathbf{x}_i, \boldsymbol{\omega})$ and corresponding labels y_i :

$$-\sum_{i=1}^N \left(y_i \log(h(\mathbf{x}_i, \boldsymbol{\omega})) + (1 - y_i) \log(1 - h(\mathbf{x}_i, \boldsymbol{\omega})) \right),$$

where N is the number of training samples, \mathbf{x}_i is the i th input training sample (image patch), $h(\cdot)$ is the network output, $\boldsymbol{\omega}$ is the set of weights of the network, and y_i is the desired output for the image patch central pixel, which will be 1 for cloud contaminated samples and 0 otherwise. In case of a patch output, i.e. when an entire patch is predicted at a time, the objective is the mean of the binary cross-entropy over all outputs. The network was trained with the Adam algorithm [13],

This work has been partially supported by the Spanish Ministry of Economy and Competitiveness (MINECO, TEC2016-77741-R, ERDF), the European Space Agency (ESA IDEAS+ research grant, CCN008), and ERC Consolidator Grant SEDAL ERC-2014-CoG 647423.

Preprint corresponding to the paper published in 2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), Fort Worth, TX, USA, pp. 2255-2258, DOI: 10.1109/IGARSS.2017.8127438.

which is a mini-batch stochastic gradient descent algorithm with adaptive estimates of lower order moments.

Finally, a common problem of classification algorithms, and of CNN in particular, is the overfitting problem that produces a poor generalization. The close relationship between the complexity of the classifier and the size of the training set suggests the idea of imposing some kind of regularization when training the models. To avoid overfitting the *dropout technique* [14] is applied at the end of both max-pooling stages (0.25 probability) and after the FC layer (0.5 probability). In addition, *data augmentation* is also employed to increase the size of the training set by adding flipped versions (left to right and up to down) of the available training samples (image patches). The data augmentation approach is shown in Fig. 1, which was previously used in the context of SVMs [15].

3. EXPERIMENTAL SETUP

Two different approaches are analyzed to apply the proposed CNN to the cloud detection of Proba-V:

- *Patch-to-pixel* classification scheme [16]: Small patches (subimages with the 4 Proba-V channels) are extracted from the image and used as input data, being the whole patch labeled according to the label of the center pixel. We test two input configurations: 4-channel 17×17 and 33×33 patches.
- *Patch-to-patch* classification scheme: Instead of classifying the center pixel we classify a central patch. Again we try 4-channel 17×17 and 33×33 patches as inputs and predict a 9×9 output patch. Figure 1 shows input and output patches as they are fed to the network.

Bigger input patch sizes make the model slower to train and to run. This is an important issue since cloud masking algorithms should be applied to all images acquired by the satellite. On the other hand, bigger input sizes allow the model to integrate more surrounding information, which could make the model more accurate. With the output patch sizes the trade-off happens the other way around: bigger output sizes will make the model faster since we predict an entire patch at a time instead of a pixel.

All CNN models were implemented in *Python* using the *Keras* library [17]. Training was done in CPUs for small models (input size 17) and GPUs for bigger ones (33×33 input). Training time ranged from 20 to 30 hours depending on the load of the computer.

In the experiments, we benchmark the proposed CNN approaches against two *state of the art* classifiers: standard fully connected neural networks (multilayer perceptron, MLP) and gradient boosting machines (GBM) [18, 19]. We trained both methods following a *pixel-to-pixel* classification scheme with different information content at the used input data: (1) the four channels of the instrument (*bands*); (2) the four channels plus ten spectral features useful for cloud detection (*feat*); and

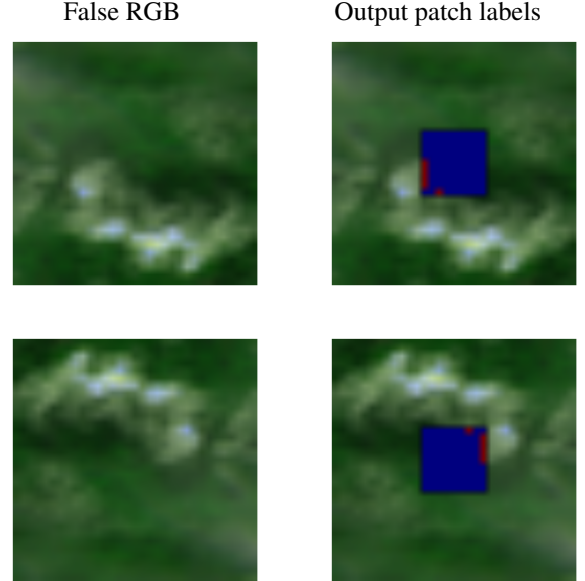


Fig. 1. Example of patch-to-patch input and output with and without data augmentation. In the first row we have the 33×33 patch and the patch with the 9×9 ground truth. In the second row the same patch is flipped and with its ground truth equally flipped

(3) the channels plus spectral features plus basic spatial features (3×3 and 5×5 mean and std) from which we finally select 40 relevant spatio-spectral inputs (denoted by *all*).

4. EXPERIMENTAL RESULTS

The available data set consists of 60 Proba-V images acquired in four days covering the four seasons: 21/03/2014, 21/06/2014, 21/09/2014, and 21/12/2014. All models are trained on 100,000 pixels randomly chosen from these images. An independent set of 360,000 pixels was left for testing purposes. The datasets were balanced to contain an equal number of cloud-contaminated and cloud-free pixels.

First, we look at the reference GBM and MLP machine learning approaches (Table 1). The effect of feature extraction results in a boost in the performance, specially when spatial information is included. Additionally, we notice that both GBM and MLP achieve similar accuracies on the test set.

In the case of CNN, first we confirm in Table 2 the improvement, due to the proposed data augmentation, on the the overall accuracy (OA%) and Cohen's Kappa statistic (κ). Then, Table 3 shows that both CNN (17×17 and 33×33 input) outperform the classical approach of *feature extraction* plus *supervised classification* when we predict the central pixel (*patch to pixel*), while *patch to patch* approaches result in lower accuracies. Figure 2 shows the accuracy over the whole 9×9 output patch, where one can observe that central pixels are more accurately predicted whereas accuracies over pixels

Table 1. Gradient boosting machines and neural networks accuracies on *pixel to pixel* classification scheme using as inputs: (1) the bands (*bands*); (2) bands and spectral features (*feat*); and (3) bands, spectral features and spatial features (*all*).

Inputs	GBM	MLP
(1) <i>bands</i>	92.92%	93.43%
(2) <i>feat</i>	93.39%	93.51%
(3) <i>all</i>	94.60%	94.51%

Table 2. Train and test set accuracy with and without data augmentation on the 33×33 *patch to pixel* CNN model. The model with data augmentation incurs in less overfitting.

Data Augmentation	Train (OA / Kappa)	Test (OA / Kappa)
NO	98.55% / 0.9709	95.05% / 0.9007
YES	96.11% / 0.9221	95.44% / 0.9085

on the boundaries of the 9×9 patches are lower.

It should be noticed that the networks trained in this work are smaller, in terms of the number of weights, than common CNN presented in the literature [16]. However, our problem can be considered less complex since there are only two classes. In addition, having a smaller network presents the advantage of a lower computational cost during the test phase. Figure 3 shows the computational time of the proposed CNN models per batch of 128 4-channel image patches. As we discussed before, smaller input sizes result in lower computational cost of the whole network. In addition, the burden of predicting a patch instead of a single value is barely noticed. Since *patch-to-patch* prediction yields a 9×9 output, patch prediction is 81 times faster than predicting the center pixel for a complete image. Nevertheless accuracy is reduced from 95.44% to 93.06% in the case of 33×33 input size (see Table 3). When choosing a 17×17 input size, accuracy loss is higher, dropping from 94.92% to 90.33%.

Finally, an illustrative example of the resulting cloud mask is shown in Fig. 4. In this figure, we show an scene of Papua New Guinea with a complex cloud structure over ocean and land. The high overall detection accuracy (95%) and Cohen’s Kappa statistic ($\kappa=0.87$) confirm the visual agreement between the cloud pattern and the obtained cloud mask.

5. CONCLUSIONS

In this paper, we presented a comprehensive study of the application of CNN models to cloud masking of Proba-V satellite images. We shown that CNN models outperform the classical approach of *feature extraction* plus *supervised classification*, even using advanced machine learning methods, in this cloud detection problem. We compared different input

Table 3. CNN accuracies of different input/output configurations. Patch to patch accuracies are measured in the center pixel and the mean over all patches overlapping this pixel.

	<i>patch to pixel</i>	<i>patch to patch</i>	
		center	mean
17×17 input	94.92%	92.90%	90.33%
33×33 input	95.44%	93.78%	93.06%

and output network configurations (patch sizes) that revealed a trade-off between classification accuracy and computational cost.

Future work is tied to better analyze the CNN training hyperparameter selection and to study the detection performance over high reflectance surfaces such as ice/snow, sand and urban areas. Another direction is to couple cloud and shadow detection on the CNN classifier.

6. REFERENCES

- [1] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton, “Imagenet classification with deep convolutional neural networks,” in *Advances in Neural Information Processing Systems* 25, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds., pp. 1097–1105. Curran Associates, Inc., 2012.
- [2] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *2016 IEEE Conf. on Computer Vision and Patt. Recognition (CVPR)*, June 2016, pp. 770–778.
- [3] Martin Längkvist, Andrey Kiselev, Marjan Alirezaie, Amy Loutfi, Xiaofeng Li, Raad A Saleh, and Prasad S Thenkabail, “Classification and segmentation of satellite orthoimagery using convolutional neural networks,” 2016.
- [4] Marco Castelluccio, Giovanni Poggi, Carlo Sansone, and Luisa Verdoliva, “Land use classification in remote sensing images by convolutional neural networks,” *CoRR*, vol. abs/1508.00092, 2015.
- [5] Fan Hu, Gui-Song Xia, Jingwen Hu, and Liangpei Zhang, “Transferring deep convolutional neural networks for the scene classification of high-resolution remote sensing imagery,” *Remote Sensing*, vol. 7, no. 11, pp. 14680, 2015.
- [6] A. Romero, C. Gatta, and G. Camps-Valls, “Unsupervised deep feature extraction for remote sensing image classification,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 3, pp. 1349–1362, March 2016.
- [7] Wouter Dierckx, Sindy Sterckx, Iskander Benhadj, Stefan Livens, Geert Duhoux, Tanja Van Achteren, Michael Francois, Karim Mellab, and Gilbert Saint, “PROBA-V mission for global vegetation monitoring: standard products and image quality,” *International Journal of Remote Sensing*, vol. 35, no. 7, pp. 2589–2614, 2014.
- [8] G. Lisens, P. Kempencers, F. Fierens, and J. Van Rensbergen, “Development of cloud, snow, and shadow masking algorithms for VEGETATION imagery,” in *IGARSS 2000. IEEE 2000 International Geoscience and Remote Sensing Symposium.*, 2000, vol. 2, pp. 834–836.

17×17 input patch to 9×9 output patch

0	87.70	88.15	88.58	89.06	89.18	89.07	88.72	88.32	87.92
1	88.07	88.53	88.92	89.50	89.78	89.51	89.13	88.81	88.47
2	88.55	88.88	89.27	89.87	90.26	90.00	89.55	89.19	88.98
3	88.91	89.44	89.99	90.93	91.68	91.07	90.13	89.67	89.39
4	89.13	89.88	90.46	91.71	92.90	91.82	90.53	90.02	89.56
5	88.86	89.36	89.83	90.75	91.42	90.83	89.89	89.68	89.29
6	88.22	88.76	89.04	89.56	89.94	89.71	89.22	89.05	88.71
7	87.75	88.24	88.56	89.03	89.27	89.15	88.79	88.41	88.12
8	87.42	87.80	88.07	88.35	88.58	88.56	88.35	87.95	87.67
	0	1	2	3	4	5	6	7	8

33×33 input patch to 9×9 output patch

0	92.28	92.50	92.53	92.68	92.89	92.79	92.70	92.82	92.42
1	92.35	92.59	92.71	92.92	92.87	92.87	92.62	92.73	92.28
2	92.35	92.66	92.87	92.87	92.93	92.76	92.85	92.65	92.05
3	92.38	92.64	92.84	93.14	93.36	93.12	92.96	92.81	92.60
4	92.55	92.78	92.96	93.40	93.78	93.51	93.11	93.17	92.99
5	92.42	92.63	92.73	93.01	93.31	93.12	92.94	92.95	92.61
6	92.24	92.60	92.64	92.86	92.94	92.81	92.65	92.73	92.49
7	92.31	92.47	92.41	92.55	92.74	92.61	92.52	92.68	92.41
8	92.45	92.41	92.34	92.49	92.60	92.63	92.47	92.59	92.34
	0	1	2	3	4	5	6	7	8

Fig. 2. Detection accuracy (%) of *patch-to-patch* models per pixel of the 9×9 output patch.

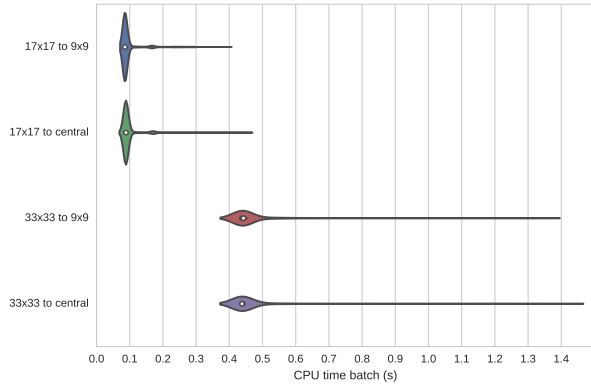


Fig. 3. CNN computational cost per batch of 128 patches using CPUs. Times measured over 2970 different batches.

RGB Composite (2014/06/21)

Predicted Cloud Mask

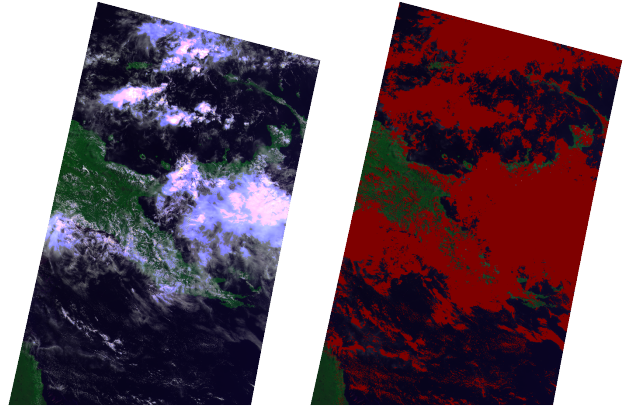


Fig. 4. Example showing the RGB false color composite and the cloud mask obtained with the 33×33 *patch to pixel* CNN.

- [9] E.L.A. Wolters, E. Swinnen, I. Benhadj, and W. Dierckx, “PROBA-V cloud detection evaluation and proposed modification,” Tech. Rep. Technical Note, 17/7/2015, QWG, 2015.
- [10] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015, Insight.
- [11] Ken Chatfield, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman, “Return of the devil in the details: Delving deep into convolutional nets,” in *Proceedings of the British Machine Vision Conference*. 2014, BMVA Press.
- [12] Sergey Ioffe and Christian Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” in *ICML*, 2015.
- [13] Diederik P. Kingma and Jimmy Ba, “Adam: A method for stochastic optimization,” in *ICLR*, 2015.
- [14] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov, “Dropout: A simple way to prevent neural networks from overfitting,” *Journal of Machine Learning Research*, vol. 15, pp. 1929–1958, 2014.
- [15] E. Izquierdo-Verdiguier, V. Laparra, L. Gómez-Chova, and G. Camps-Valls, “Encoding invariances in remote sensing image classification with svm,” *IEEE Geoscience and Remote Sensing Letters*, vol. 10, no. 5, pp. 981–985, Sep 2013.
- [16] Michael Kampffmeyer, Arnt-Borre Salberg, and Robert Jenssen, “Semantic segmentation of small objects and modeling of uncertainty in urban remote sensing images using deep convolutional neural networks,” in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2016.
- [17] François Chollet, “Keras,” <https://github.com/fchollet/keras>, 2015.
- [18] Jerome H. Friedman, “Greedy function approximation: A gradient boosting machine,” *Ann. Statist.*, vol. 29, no. 5, pp. 1189–1232, 10 2001.
- [19] Tianqi Chen and Carlos Guestrin, “Xgboost: A scalable tree boosting system,” *CoRR*, vol. abs/1603.02754, 2016.