CONVOLUTIONAL LONG SHORT-TERM MEMORY NETWORK FOR MULTITEMPORAL CLOUD DETECTION OVER LANDMARKS

Gonzalo Mateo-García, Jose E. Aduarua, Adrián Pérez-Suay, Luis Gómez-Chova

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

ABSTRACT

In this work, we propose to exploit both the temporal and spatial correlations in Earth observation satellite images through deep learning methods. In particular, the combination of a U-Net convolutional neural network together with a convolutional long short-term memory (LSTM) layer is proposed. This model is applied for cloud detection on MSG/SEVIRI image time series over selected landmarks. Implementation details are provided and our proposal is compared against a standard SVM and a U-Net without the convolutional LSTM layer but including temporal information too. Experimental results show that this combination of networks exploits both the spatial and temporal dependence and provides state-of-the-art classification results on this dataset.

Index Terms— convolutional neural networks, CNN, long short-term memory, LSTM, cloud detection, landmarks, MSG/SEVIRI.

1. INTRODUCTION

One of the key issues to further exploit information in Earth observation satellite images acquired by optical sensors is cloud contamination. Clouds are naturally present in remote sensing images (about 60% of the globe is covered by clouds). This presence of clouds can seriously hamper surface monitoring and can affect many remote sensing applications such as bio-physical parameter estimation. Moreover, most satellite missions also require cloud-free images in order to perform an accurate image co-registration. In particular, the Spinning Enhanced Visible and InfraRed Imager (SEVIRI), on board Meteosat Second Generation (MSG) geostationary satellites, provides images of the full Earth disc every 15 minutes for weather forecasts. Images in this dense time series have to be registered and this registration is based on a selection of characteristic geographic locations and landmarks over the Earth. The image registration and the geometric quality assessment in the MSG/SEVIRI products critically rely on matching these landmarks accurately. Therefore, cloud detection over landmarks is an essential step in the MSG processing chain, since undetected clouds are one of the most significant sources of error in landmark matching. Figure 1 summarizes the main steps when processing the MSG landmarks archive [1]. Once SEVIRI multispectral images are acquired, a cloud detection classifier is used to produce a cloud mask at pixel level. Then, a criterion based on the amount of cloud contamination is used to decide whether to use this landmark for the registration process or not.

![Fig. 1. MSG/SEVIRI cloud detection over Landmarks.](image)

Most cloud detection standard approaches are based on the application of thresholds on a set of discriminative spectral features. These approaches are far from being optimal and in recent years statistical machine learning methods have been shown as a powerful technique to deal with classification related tasks [2]. In particular, convolutional neural networks (CNN) have solved a broad range of problems in the computer vision field when dealing with images [3], and have been successfully applied to remote sensing and to cloud detection [4]. These networks excel in exploiting both the spectral and spatial information. However, in the particular case of cloud detection on MSG/SEVIRI data, it is also interesting to consider the sequence of images and include the temporal component, as previously done in [5]. In this work, we try to exploit the intrinsic temporal correlation between images using a given image and its immediate previous acquisitions. We combine CNN with a convolutional long short-term memory layer [6] to exploit this recursive relation between data plugging these two powerful approaches [7, 8].

The remainder of the paper is organized as follows. Section 2 explains the proposed neural network models and gives details of the selected architectures. Section 3 and 4 present the database, experiments and results. Finally, Section 5 summarizes the conclusions and future work directions.
2. METHODOLOGY

We propose to combine a convolutional neural network together with a convolutional long short-term memory layer. With the former, we seek to exploit the spatial correlations and, with the latter, we seek to capture the temporal correlations in the time series.

2.1. U-Net convolutional neural network

The CNN architecture selected for this work is the U-Net architecture presented in [9] for image segmentation. It is a fully convolutional architecture without fully connected layers, i.e. input and output image sizes are the same. This allows us to apply it to images of arbitrary size with fast inference times. This model, by itself, performs very well for the type of problem we are dealing with.

In the proposed network, we use the fully convolutional CNN as the encoding part of the convolutional LSTM layer. We also use this model as a baseline classification method to compare with.

2.2. Convolutional LSTM

To encode the existing temporal data dependencies in our model we use a LSTM layer. In particular we propose to use a convolutional LSTM (convLSTM) layer [7], which implements spatial convolutional filters instead of multiplying the stacked vector of inputs by the model weights. This allows us to keep the fully convolutional architecture of the network and, thus, the spatial structure of the images.

In a standard LSTM architecture [6] we have two types of memory: short-term, \( h_t \), and a long-term, \( c_t \). At each time step \( t \), both memories are updated using the input \( x_t \):

\[
(h_t, c_t) \leftarrow (x_t, h_{t-1}, c_{t-1}).
\]

Typically, we have three gates that control the information flow. They are the input gate \( (i_t) \), the output gate \( (o_t) \), and the forget gate \( (f_t) \). Some variations of the LSTM unit do not have some of these gates and others present extra gates. In our case, we use a modification that has an additional modulation gate \( (m_t) \).

Conceptually, the cell captures dependencies between the elements in the input sequence: \( i_t \) detects the parts of a new value that flows inside the cell while \( f_t \) controls how much the value in the cell is used to compute the output activation of the LSTM unit.

Equations for updating the memories depend on the following terms. The weights \( W \) of the connections in the LSTM are learned during the training and they determine how the gates operate. Each gate, referred by subscripts \( f, i, m, o \), is controlled by trainable weights for the input, \( W_{fx}, W_{ix}, W_{mx}, W_{ox} \in \mathbb{R}^{k \times k \times d \times r} \), and the hidden representation, \( W_{fh}, W_{ih}, W_{mh}, W_{oh} \in \mathbb{R}^{k \times k \times r \times r} \). The biases \( b_f, b_i, b_m, b_o \in \mathbb{R}^r \) are also trainable parameters. In all cases, we have that \( d \) is the input image channels, \( k \) is the convolutional kernel size, and \( r \) is the hyperparameter that determines the number of the hidden states in the recurrent layer. The sigmoid, \( \sigma(\cdot) \), and the hyperbolic tangent, \( \tanh(\cdot) \), are used as activation functions in the gates. With this, the set of equations for the convLSTMs are:

\[
\begin{align*}
    f_t &= \sigma(x_t \ast W_{fx} + h_{t-1} \ast W_{fh} + b_f) \\
    i_t &= \sigma(x_t \ast W_{ix} + h_{t-1} \ast W_{ih} + b_i) \\
    m_t &= \tanh(x_t \ast W_{mx} + h_{t-1} \ast W_{mh} + b_m) \\
    o_t &= \tanh(x_t \ast W_{ox} + h_{t-1} \ast W_{oh} + b_o) \\
    c_t &= c_{t-1} \circ f_t + i_t \circ m_t \\
    h_t &= o_t \circ c_t
\end{align*}
\]

where \( \ast \) represents convolutions and \( \circ \) represents the Hadamard product. The short-term memory \( h_t \) is used as the final output of the convLSTM network.

2.3. Proposed network topology

Our proposed network topology is shown in Fig. 2. It plugs the output of the U-Net CNN to a convLSTM layer. The input is a sequence of images. The images in the sequence are first encoded with the convolutional U-Net network. The output of this convolutional block is an image of the same size as the original one but with \( d = 32 \) channels (features). This 32-channels image is the input of the convLSTM layer. The convLSTM combines this transformed image \( (x_t) \) in the equations with its short-term \( (h_{t-1}) \) and long-term \( (c_{t-1}) \) memories. These memories are also transformed images with \( r = 64 \) channels. Finally, the short-term memory output, \( h_t \), is linearly transformed per pixel \( (1 \times 1 \text{ convolution}) \) and scaled with a sigmoid function to obtain the final cloud mask.

Fig. 2. Proposed network architecture combining the U-Net convolutional network with a convLSTM layer.
3. EXPERIMENTS

In this section we describe the data used in the experiments and the experimental setup. Then, we present the results together with a discussion.

3.1. MSG/SEVIRI landmarks dataset

We will evaluate the proposed spatio-temporal network on the problem of cloud detection for specific landmarks on Meteosat Second Generation data. MSG provides images of the full Earth disc every 15 minutes. Our dataset contains all SEVIRI Level 1.5 products acquired on the year 2010 for 200 different landmarks of variable size located over the globe (see Fig. 1). Therefore, we have a time series of 35,040 images (or chips) per landmark. Landmarks are mainly located over the coastline, islands, or inland waters. Additionally, Level 2 cloud products were available for each landmark. The Level 2 cloud mask is used as the best available ground truth to train and validate the results. It is important to note that the Level 2 cloud mask is obtained by optimal estimation methods but it is not a real ground truth, therefore, it contains errors and is biased in some regions, such as coastlines.

MSG/SEVIRI Level 1.5 products contain 12 spectral bands: 4 in the visible and near infrared range, and 8 infrared channels. In this work, we only use the channels 4, 7, 9 and 10 (i.e. infrared channels at 3.9μm, 8.7μm, 10.8μm and 12.0μm, respectively) for training the models since we are going to develop a single model for both day and night conditions. These four channels help providing temperature of clouds, land and sea surfaces.

For illustration purposes, we provide results for the Landmark located at Ad Dakhla, Morocco, Western Sahara. In our particular experience with the Landmarks dataset [1], the same techniques applied to this particular location can be extended to all the dataset providing similar behavior.

3.2. Experimental setup

In order to compare with [5], train and test datasets are split by months: for each three months of data, we use two months for training and one for validation. Therefore, as in [5], our test set comprises the months of March, June, September and December. In addition, in order to have a representative view of the accuracy along the year with a higher temporal resolution, we also train and test the networks with a weekly split: we use two out of each three weeks for training and the other one for testing.

The networks are trained to minimize a masked version of the standard binary cross-entropy classification loss. This masked version forces the network to not backpropagate the errors on the coastline pixels, where the available ground truth is less accurate. Nevertheless, those pixels are taken into account to compute the performance metrics in all test results.

We train the proposed network in two steps. First, we train the U-Net CNN without taking into account the time. Once the U-Net is trained, we use these weights as initial state of the optimization for the U-Net part of the proposed convolutional LSTM model. In the experiments, we also tried to fit the full architecture from scratch; however, the final results where only as good as the convolutional net. We trained the U-Net for 15 epochs with Adam optimizer, learning rate 0.001, and weight decay 0.05 to prevent overfitting. The convolutional LSTM was trained for 15 extra epochs with learning rate 0.005, Adam optimizer and weight decay 0.075 on the convLSTM layer.

For comparison purposes, in addition to compare with [5], we also show results of a Support Vector Machine (SVM) model proposed in [1]. This model is a combination of four independent SVM classifiers trained depending on the particular time of the day (using the acquisition solar zenith angle). These classifiers were trained using as inputs a set of spectral-spatial features, but did not use the time dimension.

4. RESULTS AND DISCUSSION

Figure 4 shows the cumulative overall accuracy for the four test months (March, Jun, September and December) for the three benchmarked models. We show in blue the SVM model, in orange the CNN model (U-Net) that does not use the temporal dimension, and in green the proposed model with the convLSTM layer. We see that the overall accuracy of the CNN model is higher than the one of the SVM model, probably because the better exploitation of the spatial information. This accuracy is also as good as the best spatio-temporal model proposed in [5]. This shows that our proposed U-Net architecture for the spectral and spatial part is better than the one used in [5]. In addition, our proposed spectral-spatio-temporal model with the convLSTM layer yields improvements of two points in accuracy against the CNN counterpart. Adding the temporal information by means of the convLSTM layer increases the performance from 87.9% in [5] to 90.4%, providing state-of-the-art results for this dataset.

Figure 5 shows the absolute overall accuracy for each test landmark chip during the whole year. In order to avoid fast oscillations of the results we used a 5-day moving average...

---

**Fig. 3.** Preview of the landmark over Ad Dakhla, Morocco: RGB composite, land-water cover mask, and coastline.
5. CONCLUSIONS

We have presented the combination of a U-Net convolutional neural network, which takes into account spatial information, together with a convolutional LSTM layer, which exploits the temporal information. The proposed methodology is applied to solve a cloud detection problem on a MSG/SEVIRI dataset consisting of a set of selected landmarks, in which both the temporal and the local spatial information are relevant and have not been properly considered in previous approaches. The proposed architecture is fully convolutional so it preserves the spatial information and structure of the image, which is even more relevant when dealing with landmark-specific local classifiers. Moreover, the temporal information from the image time series is encoded into the LSTM layer memory cells, which allows us to take into account the cloud-landmark dynamics. We show state-of-the-art results improving all previous attempts to solve this problem and open the way to test our approach over the whole landmark dataset following the same successful scheme.

6. REFERENCES


