Transferring deep learning models for cloud detection between Landsat-8 and Proba-V

Gonzalo Mateo-García, Valero Laparra, Dan López-Puigdollers, Luis Gómez-Chova

*Image Processing Laboratory, University of Valencia, Valencia, Spain

Abstract

Accurate cloud detection algorithms are mandatory to analyze the large streams of data coming from the different optical Earth observation satellites. Deep learning (DL) based cloud detection schemes provide very accurate cloud detection models. However, training these models for a given sensor requires large datasets of manually labeled samples, which are very costly or even impossible to create when the satellite has not been launched yet. In this work, we present an approach that exploits manually labeled datasets from one satellite to train deep learning models for cloud detection that can be applied (or transferred) to other satellites. We take into account the physical properties of the acquired signals and propose a simple transfer learning approach using Landsat-8 and Proba-V sensors, whose images have different but similar spatial and spectral characteristics.

Three types of experiments are conducted to demonstrate that transfer learning can work in both directions: a) from Landsat-8 to Proba-V, where we show that models trained only with Landsat-8 data produce cloud masks 5 points more accurate than the current operational Proba-V cloud masking method, b) from Proba-V to Landsat-8, where models that use only Proba-V data for training have an accuracy similar to the operational FMask in the publicly
available Biome dataset (87.79-89.77% vs 88.48%), and c) jointly from Proba-V and Landsat-8 to Proba-V, where we demonstrate that using jointly both data sources the accuracy increases 1-10 points when few Proba-V labeled images are available. These results highlight that, taking advantage of existing publicly available cloud masking labeled datasets, we can create accurate deep learning based cloud detection models for new satellites, but without the burden of collecting and labeling a large dataset of images.

Keywords: Transfer Learning, Cloud Masking, Convolutional Neural Networks, Deep learning, Multispectral sensors, Domain adaptation

1. Introduction

The number of new satellites and sensors with the objective of monitoring the Earth system and understanding its dynamics is growing exponentially. Among these sensors, optical instruments measure radiance coming from the Earth in the visible and infra-red part of the electromagnetic spectrum. Data from optical sensors is used in a wide range of applications such as estimating biophysical parameters, monitoring land use over time, assessing damages after natural disasters, or monitoring urban areas among others. In most of those applications, the presence of clouds and their shadows affects the signal and can be considered as a source of uncertainty [1]. Whereas, on a single scene, cloud masking might be handled manually, on operational applications exploiting image time series or multiple locations, this is not feasible. Thus, in order to automatically process imagery from optical sensors, accurate and automatic cloud masking algorithms are mandatory.

Cloud masking algorithms assign a **clear** or **cloudy** binary label to each of the pixels within a satellite image. Most basic approaches to cloud masking are the so called threshold based approaches, which consist of a set of thresholds applied on one or more of the spectral bands of the images, or on extracted features trying to enhance physical properties of the clouds. In general, thresholding is simple and easy to implement and it works well when the spectral information
Examples of current operational threshold based approaches to cloud masking include FMak
2 [2, 3] for Landsat-7 and Landsat-8, Sen2Cor [4] for Sentinel-2, and several recent works that propose improvements to them (e.g. 5 [6]. On the other hand, machine learning (ML) based approaches handle cloud detection as a statistical classification problem. These methods learn a cloud detection model based on a set of examples: data pairs of observations and labels. When the quality of the training data is sufficiently good, machine learning based approaches outperform threshold based ones [1, 8, 9]. Machine learning approaches to cloud detection can be further divided into classical and deep learning approaches. In classical ones, a set of manually selected spatial and spectral features are extracted for each pixel in the training set, afterwards a classifier is optimized to distinguish the label of those pixels based on these features. In the simplest case, only two classes are considered: cloud and clear pixel; however, several works consider a wider range including cirrus, cloud shadows, ice/snow, water, etc. [10, 11, 12]. Classical machine learning approaches are normally pixelwise, in the sense that the trained classifier can be applied independently to each pixel in the test image after the feature extraction. The classifiers used by these approaches widely vary including: kernel methods and support vector machines [13, 14, 15, 16], neural networks [17, 11] or trees and ensemble methods [18, 10, 19, 12]. On the other hand, deep learning approaches for cloud masking are end-to-end models where the input is the raw image and the output the cloud mask. If the model is defined as a set of stacked convolutional operations, then it constitutes a fully convolutional neural network (FCNN) [20]. In these models, the convolutional filters weights are parameters to optimize, thus the model can learn to exploit the spatial information of surrounding pixels directly from the data. FCNNs applied to cloud detection have shown state-of-the-art performance for several satellites, such as Landsat-7 [8], Landsat-8 [9, 8], GaoFen-1 [8], or MSG SEVIRI [21].

Independently of the selected cloud masking approach, the method has to be validated. This is a bottleneck in the development of cloud masking algorithms.
for most satellite sensors, since usually there is no independent and simultaneous information about the presence of clouds in the images. Therefore, in order to perform a quantitative validation, the standard approach is to manually label a set of pixels or images by human experts, which will constitute the ground truth. This approach has been extensively applied in the literature, e.g. to Landsat-7 [22], Envisat/MERIS [1], Landsat-8 [23], Proba-V [24], or Sentinel-2 [25, 26]. In some cases, only some pixels within an image are labeled as cloudy or cloud free, whereas in other cases, all the pixels of the image are labeled, capturing also the spatial distribution of clouds. In either case, this process is not exempt of errors: e.g., in [27], authors reported a mean overall error of 7% for 11 Landsat-7 scenes fully labeled by three different experts. Labeling pixels individually is more accurate, however it requires a higher dedication and hence the total amount of labeled pixels is usually considerably lower. This makes results statistically less significant which could be a problem when the goal is to validate cloud detection algorithms that work globally under different seasons and climatic conditions.

Moreover, if the proposed method for cloud detection is based on machine learning, in addition to the validation data, an independent comprehensive set of labeled samples is also required to train the models. If the goal is to provide an accurate global cloud detection method, this training set should be representative enough of natural statistics, with data from different land covers, climate zones, and seasons. Therefore, for machine learning approaches, the effort to generate a ground truth and develop a cloud detection algorithm is huge. Another disadvantage of machine learning approaches is that they cannot be applied until the satellite is launched and data is available, since a comprehensive archive of images with the corresponding ground truth is required to develop the models. For these reasons, it is still very common that most satellite missions use empirically designed threshold based methods for cloud detection at their launching time. Afterwards, if the operational cloud detection performance is an issue, the original algorithm is replaced by an improved one based on the acquired data during the mission lifetime. This is the case of Proba-V mis-
sion \cite{28}, in which case the European Space Agency (ESA) recently organized a Cloud Detection Round Robin experiment \cite{24} aimed at the inter-comparison of different cloud detection algorithms in order to improve the current operational algorithm \cite{29}.

Taking into account the aforementioned issues, we can conclude that the lack of an accurate and representative ground truth for the particular satellite sensor will hamper the development of accurate machine learning models. However, the amount of available Earth observation data is huge nowadays and it is increasingly common to publish not only the algorithms but also the manually labeled cloud masks datasets as a good practice to foster research in the field of remote sensing. In particular, for cloud detection, in \cite{23} the authors published more than 250 scenes of Landsat-7 and Landsat-8; in \cite{30} they published an additional 38 scenes for Landsat-8; the works \cite{31, 8} published 108 Gaofen-1 images and 150 high resolution scenes from Google Earth, respectively; and the works \cite{10, 32, 26} also published their manually labeled cloud masks for Sentinel-2. In this context, we propose to exploit the wealth of information contained in available labeled datasets to transfer previous knowledge about the problem between similar satellites. This approach allows us to address some of the drawbacks of machine learning approaches. Firstly, from a methodological point of view, the size of the manually labeled training set required to build an accurate cloud detection model for the new satellite is drastically reduced. Secondly, from an operational point of view, since the training data from an existing satellite is already available, the machine learning based cloud detection algorithm could be developed before the launch of the satellite, and thus it can be applied from the first day.

In this paper, we focus on the Proba-V and the Landsat-8 satellites, which have different spatial resolution, different spectral bands and from which there are manually labeled cloud detection datasets available to train and evaluate the models. Proba-V is a small satellite with medium spatial resolution and

\footnote{Upon request and for academic purpose.}
with only four spectral bands \cite{28}; we will take advantage of manually labeled
datasets for cloud detection from the recent ESA Round Robin experiment \cite{24}.
Landsat-8 \cite{33} has higher spatial and spectral resolutions compared to Proba-V,
and, as mentioned previously, there exists a large collection of manually labeled
images for cloud detection \cite{34,35}.

Our proposed approach to transfer knowledge between Landsat-8 and Proba-
V is based on two components. The first one is a domain adaptation transfor-
mation of Landsat-8 data to resemble Proba-V images in terms of both spectral
and spatial characteristics. The objective is to carry out a simple physically-
based conversion between the two sources in order to facilitate the transfer
learning from the available manually labeled dataset (i.e., the source domain)
to the satellite images where we want to detect clouds (i.e., the target domain).
The second component is a fully convolutional neural network model capable of
learning as much as possible spectral and spatial information from the training
data. FCNNs excel in image segmentation tasks \cite{36,37,38,39,40,41,42}, they
integrate spectral but also spatial information in a hierarchical manner: in our
view, spatial information is crucial specially in the context of Proba-V, which
has a limited number of spectral bands.

Using the domain adaptation transformation and the FCNN models, we
performed three types of transfer learning experiments: a) transductive transfer
learning from Landsat-8 to Proba-V, where we used only Landsat-8 annotated
data to develop a model that works in the Proba-V domain; b) transductive
transfer learning from Proba-V to Landsat-8, where labeled Proba-V data is
used to train a cloud detection model for Landsat-8; and c) inductive transfer
learning from Landsat-8 to Proba-V, where we use few Proba-V labeled images
together with the annotated Landsat-8 dataset to generate a cloud detection
model for Proba-V.

In these experiments, we show that the proposed models trained only on
Landsat-8 data (previous item a) outperforms by at least 5 points in accuracy
the current Proba-V operational cloud detection algorithm \cite{29}. This model
does not use any Proba-V image for training. In the more challenging Proba-
V to Landsat-8 transfer direction, the model trained only with Proba-V data (previous item b), which works on a 10 times lower spatial resolution scale, is only 2 point less accurate than the state-of-the-art deep learning models for Landsat-8 [9, 8] and it is as accurate as the operational FMask [2] on the analyzed dataset. Finally, the performance of models that exploit labeled data from both sensors (previous item c) shows that models trained only with few Proba-V images are significantly less accurate than models trained jointly with these few Proba-V images together with the available Landsat-8 data. In particular, results show a boost between 1 to 10 points in detection accuracy depending on the amount of Proba-V data used when networks are trained jointly using both data sources.

The paper is organized as follows. In section 2, we frame our proposal in the current literature context and we detailed our contributions. In section 3, we present the physically-based image conversion scheme, which facilitates transfer learning between sensors, the transfer learning schemes, and the proposed network architecture. In section 4, we present the Landsat-8 and Proba-V datasets. Section 5 contains the experimental design with the detailed description of the transfer learning experiments. In section 6, the results are shown and discussed. Finally, section 7 presents the conclusions.

2. Background and related work

There has been a large amount of remote sensing papers that use transfer learning in a plethora of different manners. The objective is to improve machine learning models performance by reusing data or by training the models in different but related tasks [43, 44, 45, 46, 47, 48]. Transfer learning has thus become a buzzword with different meanings depending on the particular context. In this work, we follow the definition of transfer learning given in the literature survey in [49]. In this view, the goal of transfer learning is to find a trade-off between the two most fundamental assumptions of machine learning: 1) the training set is representative enough of the underlying data distribution, and 2) future test
data is drawn from the same exact distribution. In particular, transfer learning seeks to relax this second constraint at the expense of the first one by using data from a different domain and/or from a different task when training the models.

Following this definition, transfer learning is further categorized depending on the data we have from the related domain (called source domain) and from the domain of interest (called target domain). In this work, we restrict ourselves to two of these categories: transductive transfer learning and inductive transfer learning.

Transductive transfer learning\(^{49}\) assumes that, at training time, we only have data from the source domain. This corresponds, in our setting, to use data only from Landsat-8 in order to learn a cloud detection model for Proba-V (or vice versa). This transfer learning approach is common in remote sensing when machine learning is used to invert radiative transfer models (RTM)\(^{50}\). In that case, the machine learning model is trained using simulated radiance data as input, and the variables used as inputs to RTMs as outputs. At the test phase, the machine learning model is applied to data from the target domain, which in this case corresponds to real observed satellite radiances. In the context of cloud detection, models trained on RTM simulated data has been proposed for MERIS\(^{51}\) to estimate cloud optical thickness, for MERIS and AATSR\(^{52}\), for Proba-V\(^{24}\), and also recently for MODIS\(^{53}\). The main difference with our approach is that here we transfer from a real sensor to another one. Transferring the model from a real dataset, instead that from RTM simulated radiance, has the advantage that we can exploit the natural statistics and spatial information of clouds over different surfaces, which is conveniently used by convolutional neural networks.

The second transfer learning category that we have explored is inductive transfer learning. In this setting, it is assumed to have, in addition to the source domain data, some labeled data from the target domain. Inductive transfer learning has also been explored in hybrid RTM inversions by using jointly real and RTM simulated data\(^{52, 54}\). In our setting, we will explore inductive transfer learning using all available Landsat-8 datasets and a limited number of
Proba-V real labeled images for training.

In the context of neural networks, *inductive transfer learning* is performed in at least two different ways: the first one, called *joint training*, consists of simply joining the training sets of the two domains. The second one, *fine-tuning*, consists of pre-training the network using the source domain and then use the adjusted weights as initialization for a second training using the target domain data. In the case of CNN, using the weights from ImageNet [55] as the source domain is by far the most common approach also in remote sensing applications [44, 45, 46]. This approach has been explored for cloud detection of Landsat-8 images in [8] and [56], but both works showed better performance by training a tailored fully convolutional neural network from scratch. In section 6, *joint training* and *fine-tuning* are compared experimentally.

There is a vast recent literature of deep learning applied to cloud detection on satellite imagery [57, 58, 9, 8, 59, 60, 32, 21, 30, 60]. Fully convolutional neural network is the model of choice in all those cases. In the work [8], the authors proposed a FCNN, named multi-scale convolutional feature fusion (MSCFF), for remote sensing images of different sensors, this network shows better detection accuracy than other FCNN architectures such as [57, 61]. In the work [9], authors target Landsat-8 cloud detection. They use the U-Net architecture modifying the input and output layers to accommodate for multispectral images. They designed a experimental setup where they train on the L8Biome [35] dataset and test on the L8SPARCS [34] dataset and the other way around; thus showing generalization across datasets labeled using different experts and different labeling methodologies. In this work we use the same methodology when our networks are trained and evaluated in the Landsat-8 domain.

Nevertheless, the goal of this work is not only to find an accurate FCNN architecture but to show that these FCNN models can be transferred between different sensors with very good cloud detection accuracy. Our networks are, however compared with some of these state-of-the-art methods in the Landsat-8 domain.

Finally, it is worth to mention that detection of cloud shadows is an impor-
tant issue intimately related to cloud detection. Usually, it involves two steps: first, cloud detection to locate the clouds in the image and, then, a geometry-based cloud shadow detection method [62]. This geometry-based approach is used in both Landsat-8 (Fmask) and Proba-V operational detection methods, but it could be solved also in one step in a CNN framework as shown in [54]. However, datasets including a shadow ground truth are scarcely available for Landsat [23] and are not available for Proba-V. Hence, the domain adaptation proposed in this work focuses only on cloud detection, and cloud shadows are not distinguished from other cloud-free pixels.

3. Methodology

In this section, we first introduce the Proba-V and Landsat-8 characteristics. Then, we propose two transfer learning (TL) schemes: the first of them will be used to transfer learning from Landsat-8 to Proba-V and the second one from Proba-V to Landsat-8. These TL schemes specify how training and testing can be done in the source and the target domain, respectively. Each scheme can be applied to different situations depending if the domain adaptation is done from the source to the target domain or on the opposite direction. Afterwards, we detailed the domain adaptation transformation that will be used in our experiments for both TL schemes. The transformation is based on the instrumental characteristics of the sensors in order to adapt Landsat-8 images to the Proba-V domain. Finally, subsection 3.4 explains the fully convolutional neural network architecture used in the experiments. In this paper, we focus on the Landsat-8 and Proba-V case, however, this procedure for transfer learning could be reproduced in other sensors with similar characteristics, since we only require that the two sensors have some spectral bands with overlapping response.

3.1. The Landsat-8 and the Proba-V sensors

Proba-V is a small satellite designed for global vegetation monitoring [28]. It was launched in 2013 to bridge the gap between Envisat/MERIS and SPOT
Vegetation and the recently launched Sentinel-3. Proba-V is an experimental satellite with a constrained budget designed to be much smaller than the former MERIS and SPOT. It acquires top of atmosphere (TOA) radiance in four bands of the visible (BLUE and RED), the near infrared (NIR) and short-wave infrared (SWIR). Proba-V has three cameras: one central camera, with nadir pointing, and two more on its sides. These three cameras provide a wide swath to Proba-V which enables a short revisiting period of 1-2 days. The central (nadir) camera acquires data at 100m (from 90 to 110m) whereas the spatial resolution of the two side cameras ranges from 110m to 350m. The operational Level 2A processing projects this varying resolution data into a uniform 333m Plate Carrée projection using Lanczos interpolation [63]. Cloud detection in Proba-V is specially challenging due to the limited amount of spectral information. The current operational Proba-V cloud detection algorithm based on thresholds [29] has been modified several times and it still presents several drawbacks such as a dependency on illumination and viewing geometry, the detection at edges, and the high amount of commission errors [64].

Landsat-8 [33] measures TOA radiance in 11 bands of the electromagnetic spectrum with a revisiting period of 15 days. Landsat-8 has two sensors: the Operational Land Imager (OLI), which collects data from nine spectral bands at 30m resolution; and the Thermal Infrared Sensor aimed for thermal imaging, which measures data from two more wavelengths in a 100m spatial resolution scale. There are two factors that make cloud detection an easier problem for Landsat-8 compared with Proba-V. First, the band 9 from the OLI sensor is specially designed for detection of cirrus clouds. Secondly, the thermal bands are particularly discriminative for clouds since some clouds are significantly cooler than the underlying surface. Algorithms such as FMask [3] take advantage of these facts to design simple, yet robust, cloud detection algorithms based on thresholds that can be applied globally.
### 3.2. Transfer learning schemes

As we discussed previously in section 2, transfer learning consists in exploiting data from one (source) domain to solve a problem in a similar but different (target) domain. However, there are different possibilities to perform TL depending on the relationships between the source and the target domains. In this work, we consider two different TL schemes. These schemes assume that we have labeled data in the source domain ($S$), that we want to perform predictions in the target domain ($T$), and that a domain adaptation transformation ($DA$) can be applied between both domains. The applicability of each TL scheme depends on the direction of the domain adaptation transformation:

<table>
<thead>
<tr>
<th>Scheme 1</th>
<th>Train</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>Training in Target domain</td>
<td><img src="image1.png" alt="Diagram" /></td>
<td><img src="image2.png" alt="Diagram" /></td>
</tr>
<tr>
<td>Scheme 2</td>
<td>Training in Source domain</td>
<td><img src="image3.png" alt="Diagram" /></td>
</tr>
</tbody>
</table>

Figure 1: Transfer learning schemes. In both schemes, we assume that there is labeled data in the source ($S$) domain but we want to perform predictions in the target ($T$) domain. The adaptation of images $X$ and labels $Y$ is performed using the transformations $DA_X$ and $DA_Y$, respectively. The trained ML model is denoted by $f_T$ or $f_S$, depending if it is trained in the source or the target domain, respectively.
• **Scheme 1 - training models in the target domain.** In this case we have a domain adaptation transformation from the source to the target domain. We first adapt the labeled dataset from the source domain to the target domain using the domain adaptation transformation and then we train a model using the adapted data. Since both images and labels are transformed to the target domain to train the model, applying the learned model to data from the target domain is straightforward (because the model has been constructed already in this domain). On the other hand, if we want to test the model in the source domain, we would need to transform the source domain inputs to the target domain, apply the predictive model and transform back the predictions to the source domain.

• **Scheme 2 - training models in the source domain.** This scheme is based on training the model directly in the source domain. In this scheme we have a domain adaptation transformation from the target to the source domain. In order to apply the model to new data from the target domain one has to first adapt the input sample to the source domain, then apply the predictive model, and finally transform the predictions back to the target domain. Note that, in this case, testing the model on data from the source domain is direct.

In this work, we will use Scheme 1 for transfer learning from Landsat-8 to Proba-V and Scheme 2 for transfer learning in the opposite direction. Schemes are summarized in Fig. 1. In particular, in this work, $X$ are satellite images (either Landsat or Proba-V, sec. 3.1); $Y$ are cloud mask labels; $DA_X$ represents adaptation from Landsat-8 images to Proba-V (sec. 3.3); $DA_Y$ is an adaptation of the labels by upscaling or downscaling the masks (sec. 3.3); and $f$ is the prediction model, $Y = f(X)$, implemented with Fully Convolutional Neural Networks (sec. 3.4). In addition, Figure 1 illustrates the transformations of the data to train the models and the procedure to test them in datasets from either the target or the source domains.
It is worth noting that the intended use of the proposed transfer learning models is to apply them in data from the target domain. However, it is also interesting to analyze their performance on data from the source domain since we have labeled data for validating the models and, moreover, because there are independent results from the literature to compare with. Therefore, we will evaluate the proposed models on both the source and the target domains.

3.3. Landsat-8 to Proba-V domain adaptation

In order to apply the TL schemes (sec. 3.2) we need a procedure to adapt images from one domain to the other (DA$_X$ in Fig.1). While there is vast amount of methods to learn the domain adaptation transformation from data (see e.g. [65, 66, 67]), in this paper we employ a methodology based only on the physical properties of the acquired signals. Learning the transformation would imply having data from both sensors to train a model which in some cases might not be feasible (for instance if our target satellite has not been launched yet). Hence, several parts of this study assumes there is no data (or very few) from the target domain. In cases where learning the domain adaptation transformation can be done, this simpler approach can serve as a baseline for such methods to compare with. Our proposed transformation can be applied in general when there is spectral overlap between the acquired signals in both domains. In our particular case, these domains are the Proba-V and Landsat-8 satellite images. Among the two possible domain adaptation transformations (from Landsat-8 to Proba-V or from Proba-V to Landsat-8), and given the characteristics of Landsat-8 and Proba-V images, the transformation from Landsat-8 to Proba-V seems the more natural one since it goes from the higher to the lower spatial and spectral resolutions. The opposite transformation could also be possible; however, the interpolation to a 30m spatial resolution from Proba-V images is an ill-posed problem and it is unlikely that the interpolated image has the spatio-spectral quality of a Landsat-8 image. For this reason, in all the paper, we will only consider domain adaptation from Landsat-8 to Proba-V for the transformation of TOA reflectance images.
Our proposed image conversion from Landsat-8 to Proba-V (Fig. 3) is based on the instrumental characteristics of both sensors. This conversion consists of two adaptation steps: firstly, the more suited spectral bands are selected and, secondly, we scale the Landsat-8 image to match the spatial properties of Proba-V. The spectral transformation takes into account the spectral response function (SRF) of both satellites. It consists basically in selecting the overlapping spectral bands between both satellites and eventually weight their contribution as a function of their spectral overlap. Figure 2(left) shows the spectral response function of common bands in Proba-V (solid) and Landsat-8 (dashed). One can see a good agreement in the case of the SWIR band and also in the RED one. In the case of NIR band, the spectral response of Proba-V is wider and its peak is not aligned with Landsat-8 B5 band, which might led to differences in the retrieved radiance. Finally, for the Proba-V BLUE band, there are two bands on Landsat-8 in the same spectral range. In this case, the contribution of B1 and B2 bands of Landsat-8 is weighted according to the overlapping area of the spectral responses as is shown in Fig. 2(right), which corresponds to 25% for B1 and 75% for B2.

The second adaptation step changes the spatial resolution of Landsat-8 images. In order to resemble as much as possible the spatial properties of Proba-V, we upscale the Landsat-8 image to the coarser Proba-V resolution. First, we used the point spread function (PSF) of each Proba-V spectral band to convert the Landsat-8 observations to the nominal Proba-V spatial resolution at nadir.
Figure 3: Transformation of Landsat-8 products and masks to resemble Proba-V characteristics.

The ground sampling distance (GSD) for the Proba-V center camera is about 96.9m for the BLUE, RED and NIR channels, while the SWIR center camera resolution is 184.7m (Wouter Dierckx personal communication [63], June 26, 2018). The SWIR PSF is about twice as wide as the PSF of the other bands, which stresses the fact that a distinct spatial adaptation might be applied to each band. The PSFs of the bands are modeled as 2 dimensional Gaussian filters, which are applied to the 30m resolution Landsat-8 bands. The filtered image is upsampled to the nominal 90m resolution at nadir by taking 1 out of every 3 pixels. Finally, Lanczos interpolation is applied to upscale the image to the final 333m Proba-V resolution. Notice that Lanczos is the interpolation method used at the Proba-V ground segment processing to upscale the acquired raw Proba-V data to the 333m Plate Carée grid [63].

We transformed the associated ground truth ($DA_Y$ in Fig[1]) of the Landsat-8 datasets using basically the same procedure. For the binary cloud mask, we apply the Gaussian filter, the $3 \times 3$ upscaling, and the lanczos interpolation to produce a 333m resolution image; afterwards, the image is binarized applying a threshold, which is set to 0.5 for cloudy pixels. For the transformation of
the cloud masks from the Proba-V 333m resolution to the 30m resolution, we use a simple bicubic interpolation. Spectral and spatial transformations of both Landsat-8 images and associated cloud masks are depicted in Fig. 3.

3.4. Fully Convolutional Neural Networks

Fully convolutional neural networks (FCNN) are the model of choice to learn the mapping function \( f_s \) and \( f_T \) in Fig. 1. FCNN are state-of-the-art models for image segmentation because of their capacity to exploit the spatio-spectral information of the input data. FCNNs, when provided with a large amount of training data, have shown very high accuracy levels on several image segmentation tasks [37, 38, 39]. Although the reasons of their success are still poorly understood [68, 69], it is acknowledged that the hierarchy of stacks of spatio-spectral convolutions are good priors for vision systems [70]. In addition, it has been shown in many works that they usually attain higher performance than classification methods with manually designed spatio-spectral features [12, 58].

In this work, fully convolutional neural networks solve a standard multi-output binary classification problem where the input is a 4-band image and the output is a two-dimensional map. This output has values between 0 to 1 that can be interpreted as the probability of cloud of the underlying pixel. The stacked set of convolutional filters seek to exploit the spatial information of nearby pixels to provide the cloud mask of each pixel, which is crucial in the context of reduced spectral information, with only 4 spectral bands, as in Proba-V.

Fully convolutional neural networks design has been constantly evolving since the burst of deep learning applications for image segmentation [71, 20, 72]. In most of these applications, the FCNN architectures consist of an encoder module formed of convolutional filters that pool the image several times plus a decoder module that unpool the reduced feature vectors to the original image size to conform the prediction. Since all operations are convolutions and point-wise non-linearities, the networks can be applied to images of arbitrary size with fast inference times. The U-Net architecture proposed in [73] is a well-
known fully convolutional architecture that has been applied in several fields 
from computer vision to medical imagery [73, 40, 41]. It has been extensively 
employed also in remote sensing [42, 12, 9, 21] and, in particular, for cloud 
detection with the RS-Net network [9] and in [12] for Landsat-8. It has 5 pool-
ing/unpooling stages and it adds skip connections between feature maps of the 
same resolution. Overall, the U-Net is conceptually simple yet accurate and 
provides fast predictions, which is mandatory in remote sensing and for cloud 
detection in particular. In this work, we adapted the U-Net architecture by 
reducing the number of pooling steps from five to two, by using separable con-
volutions layers [74], and by replacing the output of the network to work with 
binary classification instead of the multiclass classification. These modifications 
follow the hypothesis that cloud detection at 333m resolution can be solved with 
less parameters and with less downscaling steps. The RS-Net [9] for Landsat-8 
used 5 downscaling steps whereas we use 2, which makes sense since they were 
working with 30m resolution data.

Figure 4 shows an scheme of the proposed architecture. The encoder part 
consists of 2 blocks of two times $3 \times 3$ separable convolution, batch normaliza-
tion [75] and ReLU activation followed by a $2 \times 2$ max pooling. The bottleneck 
is also a block of two $3 \times 3$ separable convolution, batch normalization and ReLU 
activation. The decoder consists of two blocks of transpose convolution that is 
concatenated with the previous activations of the encoder, and two times $3 \times 3$ 
separable convolution, batch normalization and ReLU activation. Finally, a $1 \times 1$
convolution is applied to obtain the outputs (log-odds) that are passed through 
a sigmoid activation to obtain the final cloud probabilities. In total, our FCNN 
architecture has 95,769 trainable parameters and it does 2.18M floating point 
operations to compute the cloud mask of a $256 \times 256$ image. Compared to the 
U-Net architecture proposed in [9, 12], our proposed architecture has 99% less 
parameters and 92% less floating point operations: the U-Net has around 7.8

\[2\] The detailed implementation of the model is available at https://gist.github.com/gonzmg88/8a27dab55982817039380af1a2bf7
Figure 4: Proposed FCNN architecture, based on [12], for cloud detection: inputs are 4-band TOA reflectance images.

The model has 20 million parameters and needs 27.97M floating point operations to compute a cloud mask of a 256×256 image.

In this work, two different training strategies are used: networks are either trained from scratch and using fine-tuning. Training from scratch refers to initialize the weights of the network randomly, while fine-tuning corresponds to use the weights from a previously trained network for initialization. Since the optimization of the neural network is in general a non convex problem, a different initialization of the weights may lead to different local minimum of the loss function, which could have a different test performance.

Once weights are initialized, we used mini-batch stochastic gradient descent to minimize the standard binary cross entropy loss with respect to those weights. This loss is defined as:

\[ L(y, \hat{y}) = -\sum_{i,j,k} y_{i,j,k} \log(\hat{y}_{i,j,k}) + (1 - y_{i,j,k}) \log(1 - \hat{y}_{i,j,k}), \]

where \( \hat{y}_{i,j,k} \) is the predicted network output in the \((j, k)\) pixel of the \(i\)th image in the batch; \( y_{i,j,k} \) is its corresponding label in the ground truth; \( B \) is the batch size; and \( S_1 \times S_2 \) is the size of the image.
4. Labeled datasets

This section describes the labeled datasets used for Landsat-8 and for Proba-V. Manually annotated cloud masks are essential to train and validate cloud detection algorithms designed to work globally, over different land covers, and with different atmospheric conditions. In this work, we use the publicly available L8Biome [35] and L8SPARCS [34] datasets for Landsat-8, and an improved version of the dataset developed in the context of the ESA Round Robin exercise [24] for Proba-V (Fig. 5).

![Figure 5: Location of the used Landsat-8 and Proba-V datasets. Each image has a manually generated cloud mask.](image)

4.1. Landsat-8 datasets and ground truth

As mentioned before, one of the motivations to explore TL across Landsat-8 and Proba-V is the availability of public Landsat-8 image datasets with the corresponding cloud mask, which are used as ground truth by supervised machine learning algorithms. We use the open access L8Biome [35] and L8SPARCS [34] datasets as provided by [23].

The L8Biome dataset was developed by the authors of [23]. It contains 96 Landsat-8 Level 1T products fully labeled using three classes: clear, thin cloud,
and cloud. We fused the last two (thin cloud and cloud) to obtain a binary cloud mask. The products are scattered around the world covering the 8 major biomes. The average size of each product is 8000×8000 pixels. For some of the experiments, we used the same train-test split as in [8], containing 73 training and 19 testing images, respectively.

The L8SPARCS dataset was collected for the validation of the method proposed in [11]. It contains 80 Landsat-8 Level 1T subscenes. They were manually labeled using five different classes: cloud, cloud-shadow, snow/ice, water, flooded, and clear-sky. We merged all the non-cloud classes (cloud-shadow, snow/ice, water, flooded, and clear-sky) in the clear class for this work. Each subscene is 1000×1000 pixels, hence the amount of data compared with the L8Biome dataset is much lower.

4.2. Proba-V dataset and ground truth

The Proba-V dataset is formed by 72 Proba-V level 2A products (processing version v101) that were manually labeled by the authors. This dataset is a corrected and extended version of the dataset created in the framework of the ESA Round Robin exercise [24], which was also employed in [58, 76]. For this work, the manual labels have been extensively improved following a manual procedure by two different experts. All pixels within the 72 scenes are annotated as cloudy, clear, or uncertain. For uncertain pixels the human expert could not clearly decide whether they were cloud contaminated or cloud free. Uncertain pixels are thus not considered for neither training nor testing purposes.

In order to assess the quality of the ground truth, 950 pixels coming from 12 different images were also labeled pixel-by-pixel by independent experts [77]. The disagreement between these pixel-wise labels and the fully labeled scenes is 6.62%. This error is similar to the 7% error reported in [27]. In addition, a further analysis of the discrepancies shows that they arise mainly in semi-transparent thin clouds over the ocean, where it was difficult even for an experienced user to distinguish clouds. Nevertheless, this error constitutes a lower bound on the error a model can achieve using these labels; i.e. we cannot really
distinguish between models with errors below 6% with this dataset.

We split the Proba-V dataset into train and test. The train dataset is formed by 48 of those images: we will refer to this dataset as PV48. The test dataset is formed by the remaining 24 products, which we will call PV24. Figure 5 shows the location of the training and testing products. These labeled products are also available for inspection.\footnote{http://isp.uv.es/projects/cdc/probav_dataset.html}

5. Experimental setup

The experimental design seeks to answer several questions which can be summarized in three: 1) Can models trained with Landsat-8 data be adapted to work in the Proba-V domain? 2) Can models trained with Proba-V images (333 meter resolution) be applied to Landsat-8 images (30 meter resolution)? and 3) Does combining data from the source and target domains increase accuracy of trained models? Questions 1 and 2 are thus related to \textit{transductive TL} problems, while Question 3 involves \textit{inductive TL}. To answer these questions, we performed two blocks of experiments summarized in Table 1: one for transductive TL and one for inductive TL.

<table>
<thead>
<tr>
<th>Experiment</th>
<th>Source domain</th>
<th>Training and Testing Tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transductive TL</td>
<td>Landsat-8</td>
<td>1) Training and validation on Landsat-8</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2) Evaluation on Proba-V</td>
</tr>
<tr>
<td></td>
<td>Proba-V</td>
<td>1) Training and validation on Proba-V</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2) Evaluation on Landsat-8</td>
</tr>
<tr>
<td>Inductive TL</td>
<td>Landsat-8</td>
<td>1) Training on Landsat-8 and Proba-V</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2) Evaluation on Proba-V</td>
</tr>
</tbody>
</table>

In the transductive TL, we explored two different scenarios: 1) only having Landsat-8 labeled data and 2) only having Proba-V labeled data. For each scenario, once the models are trained, we performed two tasks: first we validate
the models in the source domain, and then we evaluate them on the target domain where they were designed to work. As explained before, in the first scenario we will use the TL Scheme 1 and in the second scenario the TL Scheme 2 (sec. 3.2).

The inductive TL experiment answers the third question formulated above. It consists of training a model in the Proba-V domain using simultaneously Proba-V data and adapted Landsat-8 data. Landsat-8 data is transformed to the Proba-V domain for training using the spatio-spectral transformation explained in sec. 3.3.

The employed TL models of each experiment are summarized in Table 2. The models are denoted by TL$_{Sat,SR}$, where Sat makes reference to the satellite from which the training data came from (L8, Landsat-8, and PV, Proba-V); SR refers to the spatial resolution used when training the model, which can be 30 meters (the Landsat-8 resolution) or 333 meters (the Proba-V resolution). Note that when the satellite is L8 and the resolution is 333m it means that, for training the model, the L8 images and the ground truth have been transformed to the Proba-V domain using the spatio-spectral domain adaptation in sec. 3.3.

<table>
<thead>
<tr>
<th>Model name</th>
<th>Source Domain (train data)</th>
<th>Target Domain (test data)</th>
<th>TL Scheme (sec. 3.2)</th>
<th>Domain Adaptation</th>
<th>TL direction</th>
</tr>
</thead>
<tbody>
<tr>
<td>TL$_{L8,30}$</td>
<td>Landsat-8 (30m)</td>
<td>Proba-V (333m)</td>
<td>Sch.1: train in target domain</td>
<td>Spectral</td>
<td>L8 to PV</td>
</tr>
<tr>
<td>TL$_{L8,333}$</td>
<td>Landsat-8 (333m)</td>
<td>Proba-V (333m)</td>
<td>Sch.1: train in target domain &amp; Spatial</td>
<td>Spectral &amp; Spatial</td>
<td>PV to L8</td>
</tr>
<tr>
<td>TL$_{PV,333}$</td>
<td>Proba-V (333m)</td>
<td>Landsat-8 (30m)</td>
<td>Sch.2: train in source domain</td>
<td>Spectral</td>
<td>L8 to PV</td>
</tr>
<tr>
<td>TL$_{L8+PV,333}$</td>
<td>Landsat-8 &amp; Proba-V (333m)</td>
<td>Proba-V (333m)</td>
<td>Sch.1: train in target domain &amp; Spatial</td>
<td>Spectral &amp; Spatial</td>
<td>L8 to PV</td>
</tr>
</tbody>
</table>
5.1. Transductive transfer learning: from Landsat-8 to Proba-V

In this experiment, we assume that we only have labeled data from Landsat-8 for training. In this setting, we trained two models that follow the TL Scheme 1. The first model, TL\textsubscript{L8,30}, uses as domain adaptation step only the spectral transformation and does not apply the spatial transformation (sec. 3.3). The second model, TL\textsubscript{L8,333}, uses both steps, the spectral and the spatial one, to adapt Landsat-8 labeled images to the Proba-V domain. Both models can be directly applied to Proba-V data. In the case of the first model, this is technically possible even though it is trained on images of different spatial resolution since the model is based on a Fully Convolutional architecture (sec. 3.4) thus it can be applied to images of any size.

In order to ensure that the models are working properly, we perform a preliminary test on Landsat-8 data. Notice that this is a realistic situation since we assume we only have labeled data from this domain. While testing the first model in the Landsat-8 domain is straightforward (i.e. the spatial resolution of the predicted cloud mask is the same as the original one), to test the second model in the Landsat-8 domain we have to undo the spatial adaptation. In order to do so, we downscale the resulting cloud mask back to the 30m resolution by simple bicubic interpolation (sec. 3.3). Specifically, we preform two tests on the Landsat-8 source domain in order to compare with the works [8] and [9]: in the first one, we follow the experimental setup of [8], which consists in using 73 images from the L8Biome dataset for training and the remaining 19 for testing\footnote{They discarded 4 images from the 96 of the L8Biome because of errors in the labels.}. The second one, following the setup of [9], uses all the L8Biome images for training and the L8SPARCS for testing.

Once we checked that the trained models work in the source domain, we evaluate their performance in the target domain (i.e. in Proba-V images from the PV24 test dataset). With this experiment we want to demonstrate that 1) transductive transfer learning works from Landsat-8 to Proba-V, and 2) that both of the domain adaptation steps (spectral and spatial) are required to enable
5.2. Transductive transfer learning: from Proba-V to Landsat-8

When assuming that we only have Proba-V labeled data for training (Proba-V is the source domain and Landsat-8 the target domain) we will apply the TL Scheme 2 (sec. 3.2). This model (TL_{PV,33} in Table 2) is first trained and evaluated in the Proba-V domain using the PV48 and the PV24 datasets, respectively. Afterwards we evaluate its performance in Landsat-8 images. To apply this model to Landsat-8 images, the TL Scheme 2 consists of 1) applying the spatio-spectral domain adaptation transformation to the Landsat-8 image (sec. 3.3), 2) applying the Proba-V trained model, and 3) downscaling the resulting cloud mask prediction back to the 30m resolution by simple bicubic interpolation.

5.3. Inductive transfer learning: from Landsat-8 to Proba-V

Finally, in the inductive block, we evaluate several models trained with an increasing amount of data coming from Proba-V and with all Landsat-8 data. Notice that, as in the first scenario of the transductive TL experiments, we use the TL Scheme 1 which trains the model in the target domain; therefore, it is straightforward to include extra labeled images from Proba-V for the joint training experiments. We analyze two different training strategies: 1) train models from scratch including simultaneously both the Landsat-8 and the Proba-V images, and 2) models are initialized using the parameters of the model TL_{L8,33} and fine-tuned with the Proba-V images. Moreover, we compare with models trained only with the same Proba-V images from scratch. The training details of all models can be found in Appendix A.

6. Experimental results and discussion

In this section, we discuss the results for the different transfer learning experiments described in section 5 and summarized in Table 1. We first present the transductive transfer learning results: we start with TL from Landsat-8 to
Proba-V (sec. 6.1), then TL from Proba-V to Landsat-8 (sec. 6.2), afterwards results related to the robustness of the transductive models (sec. 6.3), and finally a summary of all transductive transfer learning models in both domains and a comparison with independent state-of-the-art models (sec. 6.4). Finally, we present the inductive transfer learning results (sec. 6.5).

In order to test the models, we use the PV24 dataset in the Proba-V domain. In the Landsat-8 domain, we use the L8SPARCS and L8Biome datasets when they were not used for training. Testing is always performed in the native resolution of the given domain; hence, in the case of Proba-V, predicted masks are obtained at the 333m resolution domain and, in the case of Landsat-8, the predicted cloud masks are obtained at the 30m resolution of Landsat-8 images.

6.1. Transductive transfer learning results: Landsat-8 to Proba-V

In this subsection, we show results of the experimental setup explained in section 5.1. First, we show results of our models, evaluated using the same train-test split used in [8] for the L8Biome dataset, and compare our results with theirs. Then, we show results of our models trained using all images from the L8Biome dataset, which are evaluated first in the Landsat-8 domain using the L8SPARCS dataset and later in the target Proba-V domain using the PV24 test dataset. The goal of this section is to demonstrate that the transfer learning between Landsat-8 and Proba-V using the proposed spatio-spectral domain adaptation is useful. Moreover, a complementary result is that using only the spectral domain adaptation is not sufficient to obtain an accurate model.

We evaluate the models on the L8Biome dataset using the train-test split proposed in [8]. In particular, we use the same 73 images for training and 19 for testing, so results can be directly compared with [8]. We trained two models following the TL Scheme 1: the first model, TL\textsubscript{L8,30}, using as domain adaptation transformation only the spectral step and the second one, TL\textsubscript{L8,333}, using the whole spectral and spatial adaptation (cf. sec. 3.3). It is worth to emphasize that, in order to apply the models to the Landsat-8 images, the images have to be...
previously transformed using the corresponding domain adaptation transform. After the model is applied, the corresponding cloud mask has to be transformed back to the source domain. In the case of the TL$_{L8,333}$, the mask is downscaled to 30m using bicubic interpolation. Table 3 shows the results for the Landsat-8 test images. As one can see, both proposed models have a similar performance. Although the model TL$_{L8,333}$ works in a different spatial resolution, it is only one point less accurate than the model that uses directly the 30m resolution data (TL$_{L8,30}$).

Table 3: Results over the 19 test images of the L8Biome dataset used in [8]. Proposed models (TL$_{L8,333}$ and TL$_{L8,30}$) and the model from [8] (MSCFF) were all trained using the same 73 images of the L8Biome dataset.

<table>
<thead>
<tr>
<th>Model</th>
<th>Commission Error%</th>
<th>Omission Error%</th>
<th>Overall Accuracy%</th>
<th>$F_1$ score%</th>
</tr>
</thead>
<tbody>
<tr>
<td>TL$_{L8,333}$</td>
<td>6.48</td>
<td>7.67</td>
<td>92.90</td>
<td>93.11</td>
</tr>
<tr>
<td>TL$_{L8,30}$</td>
<td>6.63</td>
<td>5.58</td>
<td>93.92</td>
<td>94.17</td>
</tr>
<tr>
<td>FMask [3]</td>
<td>-</td>
<td>6.99</td>
<td>89.59</td>
<td>89.3</td>
</tr>
<tr>
<td>MSCFF [8] (all bands)</td>
<td>-</td>
<td>6.07</td>
<td>94.96</td>
<td>94.5</td>
</tr>
<tr>
<td>MSCFF [8] (NRGB)</td>
<td>-</td>
<td>5.48</td>
<td>93.94</td>
<td>92.6</td>
</tr>
</tbody>
</table>

Results from the MSCFF network [8] and from FMask [3] are included in Table 3 for comparison purposes. For the MSCFF network, we consider results using all the bands and using only the NIR, Red, Green, and Blue bands (NRGB). We can see that our network trained at 30m resolution has a similar performance than MSCFF using NRGB bands, which indicates that our FCNN architecture squeeze a similar amount of information than MSCFF even though it has much less trainable parameters and pooling steps. In addition, the network trained with the 333m resolution data (TL$_{L8,333}$) is 2 points less accurate than MSCFF using all bands [8]. However, it provides a more accurate cloud mask than the operational Landsat-8 cloud detection algorithm, FMask [3], for these 19 images. This highlights that the 333m resolution image retains sufficient information to provide an accurate cloud mask even for the 30m product;
i.e. the implicit smoothing effect of the employed upscaling-downscaling approach does not affect the overall cloud detection accuracy — although some effects at cloud borders might be expected.

Since these preliminary results were satisfactory, we retrained both networks from scratch using all the images of the L8Biome dataset as described in section 5.1. In order to analyze the robustness of the networks to different weight initialization, we trained 10 copies of the network that uses the spectro-spatial domain adaptation (TL_{L8,333}) using different random seeds. Robustness results will be further analyzed in sec. 6.3.

Table 4: Results of models trained with the L8Biome dataset and tested in the source Landsat-8 domain using L8SPARCS dataset. Both RS-Net models and our models using the L8Biome dataset for training.

<table>
<thead>
<tr>
<th>Model</th>
<th>Acc%</th>
<th>F$_1$%</th>
</tr>
</thead>
<tbody>
<tr>
<td>TL_{L8,333}</td>
<td>91.25 - 91.81</td>
<td>73.48 - 74.73</td>
</tr>
<tr>
<td>TL_{L8,30}</td>
<td>93.20</td>
<td>79.98</td>
</tr>
<tr>
<td>FMask [23]</td>
<td>92.47</td>
<td>81.61</td>
</tr>
<tr>
<td>RS-Net [9] (all-NT)</td>
<td>93.26</td>
<td>80.62</td>
</tr>
<tr>
<td>RS-Net [9] (NRGB)</td>
<td>92.53</td>
<td>76.99</td>
</tr>
</tbody>
</table>

In Table 4, the results of these models tested on the Landsat-8 L8SPARCS dataset are shown. First of all, we see that, as expected, results of the 10 copies of TL_{L8,333} exhibit a low variability for the ten different runs. This agrees with our hypothesis that a different initialization of the weights leads to consistent train and test accuracy values. Regarding the networks performance, networks trained using the spatio-spectral domain adaptation (TL_{L8,333}) are around 2 points less accurate compared with the network that work in 30m resolution (TL_{L8,30}) and the RS-Net network of the work [9]. For the RS-Net, we consider again results using the RGB bands plus NIR (NRGB) and results using all bands except the thermal (all-NT), which were the best performing model for the L8SPARCS dataset in [9]. In this case, it is also worth to mention that the network that only uses the spectral domain adaptation transformation (TL_{L8,30}) has almost the same accuracy than RS-Net [9] even tough a) the network has
99% less trainable parameters and b) it uses less Landsat-8 spectral bands.

Table 5: Results of the models trained with the L8Biome dataset and tested on the Proba-V target domain using the PV24 dataset.

<table>
<thead>
<tr>
<th>Model</th>
<th>Acc%</th>
<th>F1%</th>
</tr>
</thead>
<tbody>
<tr>
<td>TL_{L8,333}</td>
<td>88.84 - 91.87</td>
<td>87.95 - 89.71</td>
</tr>
<tr>
<td>TL_{L8,30}</td>
<td>80.37</td>
<td>73.48</td>
</tr>
<tr>
<td>Operational Proba-V v101</td>
<td>83.01</td>
<td>83.00</td>
</tr>
</tbody>
</table>

Once we showed that the proposed models trained in a transfer learning framework have a competitive performance, even with models trained specifically for the source domain, we evaluate the performance of the models in the target domain. Table 5 shows the transfer learning results of Landsat-8 models into Proba-V data. In particular, this table shows the test results of the models trained using the Landsat-8 L8SPARCS dataset and tested in the Proba-V domain using the PV24 test set. Firstly, we see that the model trained using the spatio-spectral domain adaptation (TL_{L8,333}) is much more accurate than the Operational Proba-V cloud mask. This suggests that the proposed strategy could be used to design accurate ML models even before the satellite is launched.

On the other hand, TL_{L8,333} provides results between 8 to 10 points more accurate than the model trained using only the spectral transformation (TL_{L8,30}). This demonstrates that FCNN learn spatial patterns that are dependent on the spatial resolution and, therefore, in order to transfer learning between sensors of different spatial resolutions, a domain adaptation transformation that takes into account the spatial scale is required. Results of the 10 runs of the TL_{L8,333} network show an unusual behaviour: the cloud detection accuracy and $F_1$ score vary within 3 points for the different random initializations. This dependency on the initialization contrasts with the results of these 10 runs in the Landsat-8 domain showed in Table 4. Our hypothesis is that a data-shift between the distribution of the Landsat-8 adapted data and the real Proba-V distribution still exists after the proposed adaptation. In our view, for some images in the real Proba-V domain, the networks extrapolate. Hence, predictions on these
Table 1: Discrepancies between the ground truth and three models applied to three test sites of the PV24 test dataset.

<table>
<thead>
<tr>
<th>PV False RGB</th>
<th>PV v101 Operational</th>
<th>TL_{L8,333}</th>
<th>TL_{PV,333}</th>
</tr>
</thead>
<tbody>
<tr>
<td>![Image](PV False RGB)</td>
<td>![Image](PV v101 Operational)</td>
<td><img src="TL_%7BL8,333%7D" alt="Image" /></td>
<td><img src="TL_%7BPV,333%7D" alt="Image" /></td>
</tr>
</tbody>
</table>

Cloud Mask/Ground Truth: Cloud/Cloud Land/Cloud Cloud/Land

Color Legend: ![Color Legend](Image)

Figure 6: Discrepancies between the ground truth and three models applied to three test sites of the PV24 test dataset.

regions are correct for some networks and incorrect for others depending on its initialization. However, this implicit extrapolation does not significantly affect the quality of the predictions; we can see that, even in the worse case scenario, the proposed network trained with the spatio-spectrally adapted Landsat-8 data outperforms the Proba-V operational cloud detection algorithm [29] by a large margin.
Finally, Fig. 6 shows some illustrative results of the cloud masks of three different models, all applied to Proba-V images not used for training. Those images have been selected to highlight critical cloud detection cases such as cloud ice discrimination, bright impervious surfaces, and sand and coastal areas. We show in white the agreement in cloudy pixels between the model prediction and the ground truth, in orange omission errors (the model predicts clear and the ground truth cloudy), and in blue commission errors (predictions indicate cloud and the ground truth clear). First example presents commission errors in the operational PV cloud mask over sandy beaches and water. The convolutional models do not exhibit those problems, although the model trained on the L8Biome dataset still has several omission errors mainly in cloud borders. Second example shows a winter acquisition over the Andes, in South America. In this case, the operational algorithm produce commission errors in the snowy mountains that convolutional models correctly detect; specially the model trained with Proba-V images, $\text{TL}_{PV,333}$. The last example also highlights several commission errors of the operational algorithm over the city of Istanbul, in Turkey. Again the convolutional models exhibit a much lower amount of commissions. In Appendix B, we present more examples on Proba-V images for the interested reader.

6.2. Transductive transfer learning results: Proba-V to Landsat-8

In this section, we present and analyze the results of the networks trained only with Proba-V data. These networks are first tested in the Proba-V source domain using the PV24 test set and afterwards in the target Landsat-8 domain using the L8Biome dataset. The objective is to prove that models trained in the 10 times lower resolution Proba-V data can be also transferred to the 30m Landsat-8 resolution with a negligible loss in accuracy.

We trained the model $\text{TL}_{PV,333}$ using the transfer learning Scheme 2 (Fig. 1) following the setup explained in section 5.2. In particular, we use the PV48 dataset for training and we also train 10 copies of the network to evaluate the robustness to initialization. Table 6 shows results of this model in the source
Table 6: Results of models trained in the Proba-V PV48 dataset over the Proba-V source domain using the PV24 dataset.

<table>
<thead>
<tr>
<th>Model</th>
<th>Commission Error%</th>
<th>Omission Error%</th>
<th>Overall Accuracy%</th>
<th>F1 score%</th>
</tr>
</thead>
<tbody>
<tr>
<td>TL_{PV,333}</td>
<td>4.32 - 5.61</td>
<td>4.66 - 6.01</td>
<td>94.81 - 95.10</td>
<td>94.14 - 94.43</td>
</tr>
<tr>
<td>Operational Proba-V v101</td>
<td>25.86</td>
<td>5.70</td>
<td>83.01</td>
<td>83.00</td>
</tr>
</tbody>
</table>

Proba-V domain using the PV24 test dataset. We can see that the model achieves a very high accuracy and that it is not very sensitive to the initialization of the network.

Table 7: Results of the model TL_{PV,333} over the L8Biome dataset compared with other published results. The RS-Net [9] model uses the L8SPARCS dataset for training.

<table>
<thead>
<tr>
<th>Model</th>
<th>Commission Error%</th>
<th>Omission Error%</th>
<th>Overall Accuracy%</th>
<th>F1 score%</th>
</tr>
</thead>
<tbody>
<tr>
<td>TL_{PV,333}</td>
<td>10.99 - 17.13</td>
<td>6.01 - 10.55</td>
<td>87.79 - 89.77</td>
<td>87.95 - 89.71</td>
</tr>
<tr>
<td>RS-Net [9]</td>
<td>-</td>
<td>5.51</td>
<td>91.59</td>
<td>91.52</td>
</tr>
<tr>
<td>FMask [23]</td>
<td>-</td>
<td>9.69</td>
<td>88.48</td>
<td>85.03</td>
</tr>
</tbody>
</table>

In order to test the models in the Landsat-8 domain, we follow the procedure described in section 5.2. Notice that, as we discussed before, using 333m resolution data to resolve 30m resolution images is an under-determined problem since there is information loss in the resampling process. Table 7 shows the performance metrics of the proposed model on the Landsat-8 domain using the L8Biome dataset as test set. We can see that the accuracy of the models trained with the Proba-V data, TL_{PV,333}, is similar to the accuracy of FMask [2] and it is not far from deep learning approaches of recent works [9]. This highlights that cloud detection for a given resolution can be solved reasonably well using data with lower resolution, which indicates that much of the information loss due to the upscaling does not affect the final cloud mask predictions. Appendix B presents the cloud mask of this model for some additional cherry-picked Landsat-8 images.
6.3. FCNN robustness

As previously mentioned, we trained ten copies of the same network changing the random seed for both TL directions experiments in order to test the robustness of the transductive transfer learning models to the initialization. Figure 7 shows the test accuracy of these models (TL_{PV,333} and TL_{L8,333}) in both Proba-V and Landsat-8 domains using the PV24 and the SPARCS datasets, respectively. The most clear pattern we can see is that when the networks are tested in the source domain (i.e. in the same domain that they were trained), the accuracy is higher and with lower variability than when they are tested in the target domain. As we explained before, we ascribe this behaviour to the implicit extrapolation of the networks in the target domain: the different trained networks give different predictions in some parts of the target domain that is unknown to them. These results should be taken into account when the hyper-parameters of the networks are tuned, since differences between hyper-parameters configuration might be due to noise caused by this extrapolation effect. It is also worth to mention that networks trained in the Proba-V domain (TL_{PV,333}) have similar accuracy, although with higher variability, than
the networks trained on the Landsat-8 data adapted with the spatio-spectral domain adaptation: TL_{L8,333}.

6.4. Summary of transductive results

In this subsection we explore the connections between all previous experiments and compare their results. Since the proposed models can be evaluated in both domains (Landsat-8 and Proba-V), they can be inter-compared. Table 8 shows a summary of the results from the previous sections. The first column, model, refers to a particular architecture and TL scheme employed. The TL scheme of a particular model specifies how this model is tested in the source and target domain (see sec. 5 for the details). The second column shows the dataset used to train the model; and the third column shows the dataset where the model is tested. The remainder columns are different measures of the performance of tested models. Notice that, if a given model is trained using a different dataset, it will end up with different parameters (i.e. the weight values of the network will be different).

Firstly, in the case of testing in the Proba-V domain, we can see that the model trained only with Landsat-8 is still much better than the threshold-based Proba-V Operational Cloud Detection model [29]; however, it is still far from the network trained with real Proba-V images. Therefore, it proofs to be a valid strategy with perspectives of improvement. Secondly, there is a dependency on the manual labeling procedure employed by the experts developing the ground truth: we see that models trained on data which was labeled using the same methodology for training and for testing have significantly higher accuracy. For example, networks trained on L8Biome data using the train-test split of [8] have a significantly higher accuracy (92.90%) than networks trained with all L8Biome dataset and tested in the L8SPARCS dataset (91.25-91.81%). In the case of the Proba-V domain, we see that networks trained with the PV48 dataset have also a very high accuracy in the PV24 dataset (94.81-95.10%), which may be also due to the fact that the PV48 and PV24 datasets were developed by the same experts using the same manual labeling approach. This dependence is also
Table 8: Table with results over the different test sets of the proposed models and selected models of the literature. Ranges show minimum and maximum values obtained in 10 runs changing the random seed value for the initialization of the network weights.

<table>
<thead>
<tr>
<th>Model</th>
<th>Train Set</th>
<th>Test Set</th>
<th>Commission Error%</th>
<th>Omission Error%</th>
<th>Accuracy%</th>
<th>Overall score%</th>
</tr>
</thead>
<tbody>
<tr>
<td>TL8,333</td>
<td>L8Biome</td>
<td>L8SPARCS</td>
<td>1.16 - 1.86</td>
<td>36.34 - 37.82</td>
<td>91.25 - 91.81</td>
<td>73.48 - 74.73</td>
</tr>
<tr>
<td>TL8,30</td>
<td>L8Biome</td>
<td>L8SPARCS</td>
<td>1.24</td>
<td>29.91</td>
<td>93.20</td>
<td>79.98</td>
</tr>
<tr>
<td>TL8,333</td>
<td>PV48</td>
<td>L8SPARCS</td>
<td>1.05 - 3.26</td>
<td>33.08 - 40.84</td>
<td>90.93 - 92.14</td>
<td>71.68 - 76.27</td>
</tr>
<tr>
<td>FMask [23]</td>
<td>-</td>
<td>L8SPARCS</td>
<td>-</td>
<td>13.79</td>
<td>92.47</td>
<td>81.01</td>
</tr>
<tr>
<td>RS-Net [9]</td>
<td>L8Biome</td>
<td>L8SPARCS</td>
<td>-</td>
<td>27.66</td>
<td>93.26</td>
<td>80.62</td>
</tr>
<tr>
<td>TL8,333</td>
<td>L8Biome (73)</td>
<td>L8Biome (19)</td>
<td>6.78</td>
<td>7.67</td>
<td>92.90</td>
<td>93.11</td>
</tr>
<tr>
<td>TL8,30</td>
<td>L8Biome (73)</td>
<td>L8Biome (19)</td>
<td>6.63</td>
<td>5.58</td>
<td>93.92</td>
<td>94.17</td>
</tr>
<tr>
<td>TL8,333</td>
<td>PV48</td>
<td>L8Biome (19)</td>
<td>7.32 - 10.5</td>
<td>6.83 - 9.79</td>
<td>90.85 - 91.89</td>
<td>91.11 - 92.22</td>
</tr>
<tr>
<td>FMask [23]</td>
<td>-</td>
<td>L8Biome (19)</td>
<td>-</td>
<td>6.99</td>
<td>89.59</td>
<td>89.3</td>
</tr>
<tr>
<td>MSCFF [8] (all bands)</td>
<td>L8Biome (73)</td>
<td>L8Biome (19)</td>
<td>-</td>
<td>6.07</td>
<td>94.96</td>
<td>94.5</td>
</tr>
<tr>
<td>MSCFF [8] (NRGB)</td>
<td>L8Biome (73)</td>
<td>L8Biome (19)</td>
<td>-</td>
<td>5.48</td>
<td>93.94</td>
<td>92.6</td>
</tr>
<tr>
<td>TL8,333</td>
<td>PV48</td>
<td>L8Biome</td>
<td>10.99 - 17.13</td>
<td>6.01 - 10.55</td>
<td>87.79 - 89.77</td>
<td>87.95 - 89.71</td>
</tr>
<tr>
<td>FMask [23]</td>
<td>-</td>
<td>L8Biome</td>
<td>-</td>
<td>9.69</td>
<td>88.48</td>
<td>85.03</td>
</tr>
<tr>
<td>RS-Net [9]</td>
<td>L8SPARCS</td>
<td>L8Biome</td>
<td>-</td>
<td>5.51</td>
<td>91.59</td>
<td>91.52</td>
</tr>
<tr>
<td>TL8,333</td>
<td>L8Biome</td>
<td>PV24</td>
<td>5.10 - 12.18</td>
<td>8.42 - 14.63</td>
<td>88.84 - 91.87</td>
<td>87.20 - 90.69</td>
</tr>
<tr>
<td>TL8,30</td>
<td>L8Biome</td>
<td>PV24</td>
<td>5.00</td>
<td>38.23</td>
<td>80.37</td>
<td>73.48</td>
</tr>
<tr>
<td>TL8,333</td>
<td>PV48</td>
<td>PV24</td>
<td>4.32 - 5.61</td>
<td>4.66 - 6.01</td>
<td>94.81 - 95.10</td>
<td>94.14 - 94.43</td>
</tr>
<tr>
<td>Oper. PV v101 [29]</td>
<td>-</td>
<td>PV24</td>
<td>25.86</td>
<td>5.70</td>
<td>83.01</td>
<td>83.00</td>
</tr>
</tbody>
</table>

Documented in other contexts involving classification like in [79, 78]. In the case of cloud detection, this could be exacerbated by different criteria in the inclusion of thin clouds in the datasets, since in one dataset very thin semitransparent clouds might have been considered as clear pixels whereas for other this pixels might have been annotated as cloudy. Finally, in these results, it is important to consider the errors in the labeling procedure. These errors were estimated to be around 7% for Landsat [27] and 6.62% for Proba-V (see section 4.2). Hence, models over 93% accuracy cannot be really compared or ranked, from a statistical point of view, using these datasets.

6.5. Inductive transfer learning results

In this section, we present and discuss results of models that use both datasets for training, simultaneously. This setting seeks to explore a scenario where there are few labeled images from a given (target) satellite sensor, which
is often the case due to the high cost of manual labeling of clouds, and a larger corpus of labeled images from a different but similar sensor. Proba-V will be in these experiments the target domain, where few labeled images with cloud mask are available, whereas the Landsat-8 satellite will be the source domain with the L8Biome dataset as the large corpus of labeled images. The goal of the experiments is thus to test if networks trained using fine-tuning or joint training with the L8Biome dataset have a significantly better performance than networks trained from scratch using the few Proba-V images.

In order to train the models with Landsat-8 data we apply the TL Scheme 1 (sec. 3.2) with the proposed spectro-spatial domain adaptation as explained in section 5.3. In this setting, models are trained in the Proba-V domain hence, joint training consists of merging the dataset of the few Proba-V images with the dataset of Landsat-8 adapted images.

We trained several networks with an increasing number of real Proba-V images \(d\) from the PV48 dataset. For each number of Proba-V images, \(d\), we selected 8 disjoint subsets of the PV48 dataset containing \(d\) images. For each of such subsets, three models were trained: a) from scratch using the \(d\) Proba-V images in the subset; b) fine-tuning, which uses those \(d\) Proba-V images to fine-tune a network trained previously in the L8Biome dataset\(^5\); and c) joint training, which trains from scratch using the \(d\) Proba-V images together with all the images in the L8Biome dataset.

Figure 8 shows the results of this experiment tested over the PV24 test set. Overall, we see that joint training has a better performance than training from scratch or using fine-tuning, which shows similar accuracy. In particular, we can see that using joint training increases the mean accuracy between 2-4 points in the scarce data scenarios with 1 to 3 images. In scenarios with 4 to 6 images for training, joint training still gives an small boost in accuracy and also reduces the variance of the resulting accuracy values. This indicates more robustness of the joint training solution. In scenarios with a larger amount of data, we see that

\(^5\)For fine-tuning we used the network TL\(_{\text{L8,333}}\) from subsection 6.1
the three methods have a similar performance. It is also worth to mention that joint training provides systematically an increase in the mean accuracy over the models trained only with Landsat-8 data without any Proba-V image (sec. 6.1).

Figure 8: Test accuracy over the PV24 test set of FCNN Joint models trained with different numbers of Proba-V images in red from scratch, in yellow using fine-tuning and in green using joint training.

Figure 9 compares joint training with training from scratch. In this figure, each point represent a subset of $d$ images with $d$ varying in the x-axis. Points on the left show the accuracy on the PV24 test set of the model trained from scratch using only those $d$ images, whereas points on the right use these $d$ images together with the images in the L8Biome dataset for training (joint learning). We see that, for the vast majority of those subsets, using joint training has a positive impact on the final performance of the model (points on the right). We see again that the variance of the joint models is reduced and that joint training consistently take advantage of the new Proba-V data to also perform better than the model that does not use it, which is trained only with the L8Biome dataset and depicted by the blue shaded area. Note that the orange area depicts the accuracy of models trained on all Proba-V images (PV48) that provides an upper bound for the cloud detection accuracy.
Figure 9: Test accuracy of models trained using different number of Proba-V images. For each value, on left, only Proba-V data is used; on right, models trained jointly on Landsat-8 and Proba-V data. Blue shaded area depicts the accuracy of the models trained only in the L8Biome dataset. Orange area depicts the accuracy of models trained on all Proba-V images (PV48).

7. Conclusions

In this paper, we explored different transfer learning (TL) approaches to train machine learning (ML) methods for cloud detection in remote sensing images. In particular, we analyzed transductive and inductive TL frameworks using Landsat-8 and Proba-V as case studies. Both frameworks depend on a domain adaptation transformation that converts images from one satellite to resemble images acquired with the other satellite.

We proposed an image conversion method to adapt Landsat-8 images to the Proba-V spectral and spatial characteristics that enables TL across satellites. Our results suggest that it is important to use both the spatial and the spectral adaptation in order to fully exploit TL advantages.

The transductive transfer learning framework assumes that we only have data from one satellite. In this context, two different TL schemes were proposed and successfully tested. Each scheme allows for a different TL depending on the particular direction of the domain adaptation transformation: from the source domain to the target domain or from the target domain to the source domain.
We show that ML models trained only with data from Landsat-8 can have a very good performance on Proba-V surpassing current operational algorithm [29]. This means that ML methods can be trained even before the satellite is launched, and obtain better performance than threshold-based approaches. We evaluated the proposed methods results in the context of state-of-the-art cloud detection methodologies based on deep learning [9] and [8].

In order to use the Proba-V data for predicting on Landsat-8 images, we proposed a TL scheme that takes advantage of the proposed Landsat-8 to Proba-V domain adaptation transformation. We showed that ML models trained only with Proba-V data have similar accuracy than operational Landsat-8 approaches such as FMask [2] and are only two points less accurate than [8], even though our method is trained with data on a 11 times lower spatial resolution.

The inductive transfer learning framework relies on merging data from two different domains. We showed that joining data from both satellites increases accuracy specially in the regimes where there is few data from the target Proba-V domain, although we do not see a significant improvement using fine-tuning.

We show that training only with the adapted Landsat-8 data suffers the data-shift [78] problem. In particular, we trained 10 copies of the same network with different initialization weights and show that the error in the adapted Landsat-8 domain is lower and with less variance than in the Proba-V domain. We see that, in the former, the error ranges from 91.4% to 91.9% whereas for Proba-V the error is 88.8-90.7%. This contrasts with the belief that CNN initialization does not affect much the obtained solution. In this respect, there is still margin to improve the transfer learning results by improving the domain adaptation transformation and thus reducing the data-shift problem. Our next steps are fostered to improve the cloud detection accuracy by using the generative adversarial networks (GANs) framework [80] to learn a transformation between Landsat-8 and Proba-V data.
Acknowledgements

This work has been partially supported by the Spanish Ministry of Science, Innovation and Universities (MINECO, TEC2016-77741-R, ERDF), the European Social Fund and the European Space Agency (ESA IDEAS+ research grant, CCN008). We also thank the Proba-V Quality Working Group members and the Proba-V Cloud Detection Round Robin exercise participants, with a special mention to Wouter Dierckx (VITO) and Fabrizio Niro (ESA).

References


URL https://earth.esa.int/documents/700255/2362868/ProbaV_CloudContest_ValidationReport_1_3.pdf


Appendix A. Models training technical details

For convenience, all networks were trained using as input patches of size 32×32, although the model is independent of this size. The test size corresponds to the size of the images: for example, in the case of the L8SPARCS dataset, the size of each test image is 1000×1000 pixels. The input of the networks is always top of atmosphere reflectance for both Proba-V [81] and Landsat-8 [82]. The training patches were taken from the training images with a 16 pixel overlap as a form of data augmentation; we also employed horizontal and vertical flips and 90 degree rotations as data augmentation techniques. In all experiments, we used 64 as the batch size and the Adam optimizer [83]. All networks were trained using TensorFlow library (v1.12) [84] and the weights of the network were initialized with the default initialization of each of the corresponding layers.

The networks TL_{L8,333} are trained in Landsat-8 data from the L8Biome dataset transformed using the spatio-spectral domain adaptation transformation. After the spatial domain adaptation, the L8Biome dataset has 243,430


patches. We first train the network using the train-test split of [8] to compare with their approach. Then, the network is trained using all the L8Biome dataset. Networks were trained until no improvement was observed for 15 epochs. We used a learning rate of $10^{-4}$ and weight decay of $5 \cdot 10^{-3}$ for regularization purposes. We trained 10 copies of the same network architecture with different random seed initialization to ensure that results do not depend on the weight initialization or optimization process.

Networks $\text{TL}_{\text{PV},333}$ are trained on the Proba-V data using the PV48 Proba-V dataset which corresponds to 1,891,095 patches. For these experiments we reduced the learning rate and the weight decay. In particular, we used a learning rate of $10^{-5}$ and weight decay of $5 \cdot 10^{-4}$. We also trained 10 copies of the same network to ensure consistency across initializations.

Networks $\text{TL}_{\text{L8},30}$ are trained in the L8Biome dataset transformed using only the spectral domain adaptation. Since there is no spatial upscaling, the total amount of patches is much bigger than in the $\text{TL}_{\text{L8},333}$ case (14,531,228 patches). We also trained the network first using the train-test split of [8] to compare with their approach. Networks were trained for 50 epochs using a learning rate of $10^{-5}$ and weight decay of $5 \cdot 10^{-4}$.

When the networks are trained jointly, we used the L8Biome dataset transformed using the spatio-spectral domain adaptation (i.e. same as above in $\text{TL}_{\text{L8},333}$), and an increasing number of images from the PV48 dataset. For the fine-tuning we used as initial weights the aforementioned network ($\text{TL}_{\text{L8},333}$) trained with the L8Biome dataset.

**Appendix B. Visual inspection of cloud detection results**

In this appendix, we show additional results of the produced cloud masks for Proba-V and for Landsat-8, and compare them against the ground truth. All shown images have not been used for training by none of the models. As in section [6], we show in white agreement in cloudy pixels between the model prediction and the ground truth, in orange omission errors (the model predicts
clear and the ground truth cloudy), and in blue commission errors (predictions indicate cloud and the ground truth clear).

Figure B.10 shows four additional results for Proba-V. Models shown are the operational Proba-V cloud detection [29], the model trained on Landsat-8 data with the proposed domain adaptation $TL_{L8,333}$ and the model trained on Proba-V $TL_{PV,333}$. The first example presents omission errors of the operational Proba-V algorithm over cloudy areas with saturated pixels in the blue band. We see that both convolutional models solve this issue, nevertheless, the model trained on the L8Biome dataset still has several omission errors in cloud borders.

Second example shows an acquisition over Corsica island, where all models capture most of the cloudy pixels. However, in this case, the operational model has commission errors in the snowy mountains in Corsica that convolutional models correct; specially the FCNN trained with Proba-V images $TL_{PV,333}$. The third example also highlights several commission errors of the operational algorithm, in this case over coastal waters. Again the convolutional models do not exhibit this problem although there are very thin clouds over land that are undetected. Finally, last acquisition shows a salty lake in Central Anatolia (Tuz Lake), where we can see that the operational algorithm incurs in several commission errors. The models based on FCNNs exhibit less commissions in the case of $TL_{L8,333}$ and none in the model trained with Proba-V data $TL_{PV,333}$.

Figure B.11 shows results for Landsat-8 of four images in the L8SPARCS dataset. In the case of Landsat-8, the models selected are the operational FMask [3], the model trained with Proba-V data $TL_{PV,333}$ and the model trained with Landsat-8 data at its original resolution $TL_{L8,30}$. First row shows the Vichada river in the border of Colombia and Venezuela. We can see that overall all three models provide sensible cloud masks. FMask exhibit a slightly higher amount of omission errors for very thin clouds in the bottom part of the images which the models based on FCNN do not exhibit. Second row shows the tundra in the North of Quebec in late spring. We can see several commission errors of FMask in regions where the ice is melting; these false positives are not present in the FCNN models predictions nevertheless the model trained in
Cloud Mask/Ground Truth: Cloud/Cloud  Land/Cloud  Cloud/Land
Color Legend:  

Figure B.10: Discrepancies between the ground truth and three models applied to four test sites of Proba-V.
Proba-V data omits some clouds in the icy surface. Third row, from the chilean coast in South America, shows very thin clouds in the upper right part of the images that the three models mainly omit. In addition, FMask shows systematic commission errors in the coast pixels that the FCNN models do not have. Last row is an acquisition from a salt marsh in the Little Rann of Kutch in India. It contains muddy water with a big amount of suspended sediments and salt evaporation ponds. We can see that FMask has large commission errors in these muddy waters and it also failed to identify thin clouds in the bottom right of the image. In contrast, the model trained on Proba-V data shows few commission errors mostly in the salt pans and it is able to capture most of thin clouds in the image. The model trained with 30m Landsat-8 data does not show commission errors, however, it also failed to identify several thin clouds.
Cloud Mask/Ground Truth: Cloud/Cloud  Land/Cloud  Cloud/Land

Color Legend:

Figure B.11: Discrepancies between the ground truth and three models applied to four test sites of the L8SPARCS dataset.