Journal of Applied Remote Sensing

RemoteSensing.SPIEDigitalLibrary.org

Deep convolutional neural networks for land-cover classification with Sentinel-2 images

Eleni Kroupi
Maria Kesa
Victor Diego Navarro-Sánchez
Salman Saeed
Camille Pelloquin
Bahaa Alhaddad
Laura Moreno
Aureli Soria-Frisch
Giulio Ruffini

SPIE.

Eleni Kroupi, Maria Kesa, Victor Diego Navarro-Sánchez, Salman Saeed, Camille Pelloquin, Bahaa Alhaddad, Laura Moreno, Aureli Soria-Frisch, Giulio Ruffini, "Deep convolutional neural networks for land-cover classification with Sentinel-2 images," *J. Appl. Remote Sens.* **13**(2), 024525 (2019), doi: 10.1117/1.JRS.13.024525.

Deep convolutional neural networks for land-cover classification with Sentinel-2 images

Eleni Kroupi,[a,*] Maria Kesa,[b] Victor Diego Navarro-Sánchez,[b] Salman Saeed,[b] Camille Pelloquin,[b] Bahaa Alhaddad,[b] Laura Moreno,[b] Aureli Soria-Frisch,[a] and Giulio Ruffini[a,b]

[a]Starlab SL, Department of Neuroscience, Barcelona, Spain
[b]Starlab SL, Department of Space, Barcelona, Spain

Abstract. Currently, analyzing satellite images requires an unsustainable amount of manual labor. Semiautomatic solutions for land-cover classification of satellite images entail the incorporation of expert knowledge. To increase the scalability of the built solutions, methods that automate image processing and analysis pipelines are required. Recently, deep learning (DL) models have been applied to challenging vision problems with great success. We expect that the use of DL models will soon outperform shallow networks and other classification algorithms, as recently achieved in multiple domains. Here, we consider the task of land-cover classification of satellite images. This seems particularly appropriate for deep classifiers due to the combined high dimensionality of the data with the presence of compositional dependencies between pixels, which can be used to characterize a particular class. We develop a pipeline for analyzing satellite images using a deep convolutional neural network for practical applications. We present its successful application for land-cover classification, where it achieves 86% classification accuracy on unseen raw images. [DOI: 10.1117/1.JRS.13.024525]

Keywords: deep learning; remote sensing; land-cover classification; Sentinel-2.

Paper 180970 received Dec. 9, 2018; accepted for publication May 24, 2019; published online Jun. 20, 2019.

1 Introduction

Baseline urban classification (BUC) and change maps are Earth observation products with a high value for post analysis of urban tissue and associated urban planning activities. Currently, the semiautomatic methods providing such products require the intervention of a qualified professional for postvalidation and updates. The need to establish valid BUC maps is crucial for tracking land use and land-cover changes both in rural and in urban environments. Such monitoring serves for a variety of causes, including urban planning, land quality, and land management, which may influence political decisions, environmental protection, and economics. The objective of the present work is to develop an innovative classification framework based on deep learning (DL) to produce the same products with, at least, the same accuracy, through a fully automatic process. Below, we review the products associated with the current land-cover classification, the current method and its limitations, and the new potential methods that are expected to produce more scalable systems from the use of emerging machine learning algorithms.

2 Literature Review

2.1 *Baseline Urban Classification Maps*

The BUC maps provide geolocated visual data of urban land use such as artificial surfaces, nonartificial surfaces, and other natural and seminatural areas (Fig. 1). The broadest level of

*Address all correspondence to Eleni Kroupi, E-mail: eleni.kroupi@starlab.es

1931-3195/2019/$25.00 © 2019 SPIE

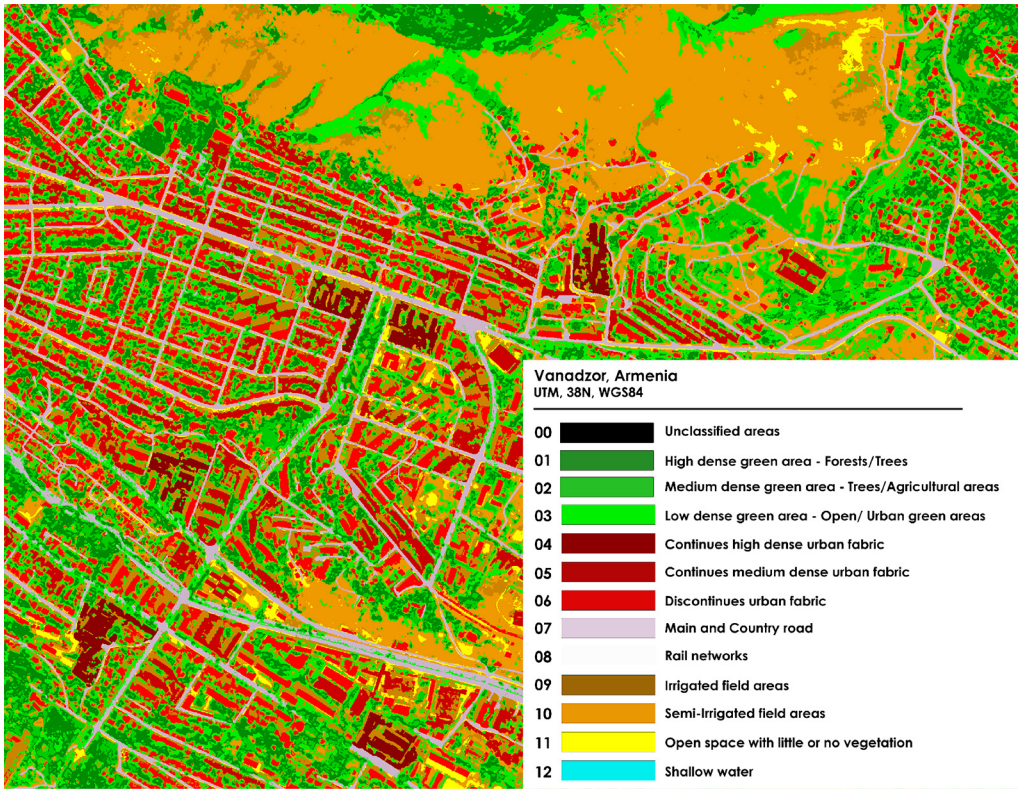


Fig. 1 BUC Vanadzor, Armenia, from the ESA SCUDA project. 1.2 English.

categorization (level I) distinguishes among land-cover types: urban, agricultural, forest, water, irrigated lands, etc. For urban land, the second level of categorization (level II) distinguishes among thematically detailed land uses: high and medium, dense and discontinuous urban fabric, main and country road, and rail network.

BUC maps can be further processed and combined to provide land use change (LUC) maps over two points in time (2002 and 2014), which detail the spatial characteristics of settlement evolution within the areas of interest.

However, the semiautomatic solutions currently provided for BUC maps entail the incorporation of qualified professionals for manual correction and validation of the land-cover results, as currently there are no sufficiently accurate automatic methods to achieve this. The appearance and superior performance of deep neural networks (DNNs) over other conventional classifiers on other image segmentation tasks (e.g., Refs. 1–3) make DNNs look promising also for BUC classification.

2.2 *Conventional Baseline Urban Classification Methods*

One of the most used approaches for BUC classification is region-based segmentation, which is implemented through several stages. First, the image is automatically segmented into regions that fulfill homogeneity criteria. Second, some segments are assigned to classes manually. Last, these manually labeled data are used as training set for different algorithms, i.e., K-nearest neighbor (KNN) or support vector machine (SVM) that classify the pixels of the entire image (see Ref. 4, for a review on these methods). It is not uncommon to merge the results from several classifiers based on their estimated performance over different sets of land uses. Moreover, supervised classification outputs are generally further refined through morphological operations, texture-driven filters, and other image-processing techniques in order to improve the final accuracy, usually relying on expert criteria on what sort of postprocessing is required. As an example, such an approach has been used to generate BUC maps in the context of ESA's SCUDA project using imagery from Pleiades (very high resolution, 0.5 to 1 m) and Spot-5&6 (high resolution,

5 to 10 m). Even if the maps retrieved from Pleiades data have shown a high accuracy overall, the maps obtained from Spot imagery following the described approach have provided lower accuracies, especially for built-up areas. For the specific study area of Vanadzor (Fig. 1), the urban class reported 95.69% producer's accuracy (PA) and 32.84% user's accuracy (UA), which suggests that a different approach is required to improve performance when resolution is limited.

Other methods often used include unsupervised and semisupervised techniques (e.g., Refs. 4 and 5). In this context, it is worth pointing out that the classification methods commonly used in image analysis practice typically present a lower performance than modern methods based on DL (e.g., in machine vision competitions, such as the eminent ImageNet,[6] the top performers are DL algorithms). We, therefore, expect that adopting DL in the image analysis pipeline will significantly improve the obtained accuracies while decreasing the need for manual labeling of segments in the second stage.

2.3 *Deep Learning Methods for Satellite Images*

Recently, DL models have been applied to challenging vision problems with great success,[7,8] and the application of DL models outperforms shallow networks and other classification algorithms, as recently achieved by DL approaches on satellite images.[9–12] For instance, Basu and colleagues[9] use deep belief networks with feature engineering for satellite image classification. On the SAT-4 dataset the best network produces a classification accuracy of 97.9% on 500,000 image patches covering four broad land-cover classes. It produces 11% better classification accuracy than state-of-the-art object recognition algorithms, convolutional neural networks (CNNs) and stacked denoising autoencoders.[13] Deep convolutional neural networks (DCNNs) also outperform other commonly used classification approaches in remote sensing, such as the random forests and the traditional fully connected multilayer perceptron, on multitemporal scenes acquired by the Landsat-8 and Sentinel-1A satellites.[14] The DCNNs yield 85% accuracy for all major crops. Other works using DL methods on very high spatial resolution image classification include images acquired from the Quickbird II instrument,[15] as well as fusion of hyperspectral and PolSAR data using a DCNN that outperforms conventional fusion with an SVM classifier.[16] Sparse stacked autoencoders have been also used to learn an effective representation of the input data (e.g., Refs. 17–20), as well as recurrent neural networks and supervised and unsupervised CNNs (e.g., Refs. 14–16, 21–27). The currently used DL methods and their advantages in remote sensing applications are extensively described in a review paper and a technical tutorial.[28,29]

However, although a variety of DL methods have been recently applied to various satellite image datasets, the majority has been applied to high or very high resolution images (e.g., Refs. 30–34). Research closer to our work has been mainly carried out by Refs. 35 and 36.

Another trend in the land-cover classification procedure for satellite images is the use of semisupervised learning approaches, which have been applied with excellent results.[37,38] This methodology allows reducing the labeling time of data from ~9 h to 30 min. The semisupervised approach achieves 87.9% compared to 91.5% (Sydney dataset) and 86.8% compared to 92.5% (Washington dataset) fully supervised model accuracy. This method uses discriminative sparse autoencoders for extracting high-level features from data. The literature pertaining to the application of semisupervised DL algorithms with satellite images is sparse, perhaps because of the relatively recent emergence of this approach and its technical challenges, and also because deep networks require a large amount of training data, which is usually not the case in semisupervised approaches that require artificially generated data to compensate. There is some older work which employed neural networks (see Refs. 39 and 40), but modern DL approaches are only beginning to be explored by the community.

Given that the above results using DL algorithms are sufficient for ensuring the corresponding operational application, our aim is to increase the classification accuracy also for other types of satellite images, e.g., in the Sentinel-2 dataset. We will minimize in this manner the human labor involvement in the classification chain also for this type of data. As the Sentinel-2 satellite is one of the most recent high resolution ones, to our knowledge there are currently only few studies on DL from Sentinel-2 data. One of the problems that we need to deal with in the industrial sector when using Sentinel-2 data is the absence of ground truth (GT) in order to develop

and train our algorithms, so that we later directly apply them on data from other countries/continents of the same satellite.

To achieve this goal, we need first to evaluate the performance of the DCNNs trained on an already-existing database of Sentinel-2 satellite images that contains GT, and test it on unseen images of another continent and of the same satellite type. Having a model already trained and being able to directly recall it to classify with a good accuracy new images of the same satellite type will already reduce the time spent in land-cover classification in a semiautomated way. The DCNNs are applied in this study, as there has not been a gold standard of land-cover conventional classification methods so far, and the deep networks seem to outperform conventional machine learning algorithms in various image classification problems (e.g., Refs. 1–3). The image used as the test set has been previously classified in a semiautomated manner using the semantic segmentation approach, which is described in the former section. Our current aim here is to achieve good classification performance with these images by training a DCNN with an existing database and applying the model to the new image. This paper is an extension of our previous work,[34] having added models that provide finer resolution and comparing them, visual examples and evaluations of the best performing model as well as extended the relevant discussion part, and a section that deals with the controversial pixels as detailed in Sec. 3.4.

3 Proposed Methods and Materials

3.1 *Deep Learning Scheme*

Our images and the problem we aim to solve are derived from the Sentinel-2 high-resolution multispectral satellite imager, which collects 13 spectral bands. However, to simplify the problem, in this study we restrict the analysis to four bands, namely RGB and near-infrared (NIR). The procedure followed is summarized in the following steps: (1) initially, an architecture similar to a model that has been shown to provide very high classification accuracies in the DeepSat database is implemented in TensorFlow following[41] (please note that the words architecture and model are used interchangeably in this text). In Ref. 41, the authors proposed an architecture that outperformed pretrained conventional networks (such as the AlexNet), and, thus, in our work we have implemented a similar architecture. Briefly, the model contains seven types of layers, namely convolutional layer, rectified linear unit layer, maxpool layer, dropout layer, a threshold layer, fully connected layers, and a final softmax layer. The layers are arranged sequentially. The only change made with respect to the initial model is the use of the Adam optimizer that already includes learning rate decay, so the exponential learning rate decay used in Ref. 41 has not been implemented. (2) Once the model is implemented, it is initially trained and tested in the Modified National Institute of Standards and Technology (MNIST) database,[42] and (3) later in the DeepSat database, in both cases for validation purposes. The results are presented in Sec. 4.

The initial DCNN model as described in Ref. 41 used 28×28 pixel patches as inputs. (4) When training with the EuroSat database (see Sec. 3.2), we used the same architecture but with 32×32 pixel patches and 16×16 pixel patches as inputs, and (5) adapted the architecture as explained below for 8×8 and 4×4 pixel patches of the same dataset. The network architecture was changed for these cases for the following reason. Using the original network in the 8×8 case, the first convolutional layer with stride 1 and valid padding decreased the dimensionality of the 8×8 image to 6×6. The first maxpool layer with kernel size 2 further decreased it into 3×3. The second maxpool layer with kernel size 3×3 further downsampled the image into a 1×1 array. Thus, the next maxpool layer with kernel size 2×2 can no longer downsample the output 1×1 array. This points out the need of decreasing the intermediate layers so that the network can be adapted size-wise to smaller initial patches. Thus, to convert the network to a suitable model for 8×8 classification, we removed the fifth convolutional layer and the third maxpool layer leading to a final network, as presented in Fig. 2(b). Following a similar rationale for the 4×4 case we removed the third, fourth, and fifth convolutional layers, as well as the second and third maxpool layers, leading to a network architecture, as in Fig. 2(a).

The rationale behind decreasing the patch size was to get the final classification result in a finer resolution without losing the information of neighboring pixels provided by the

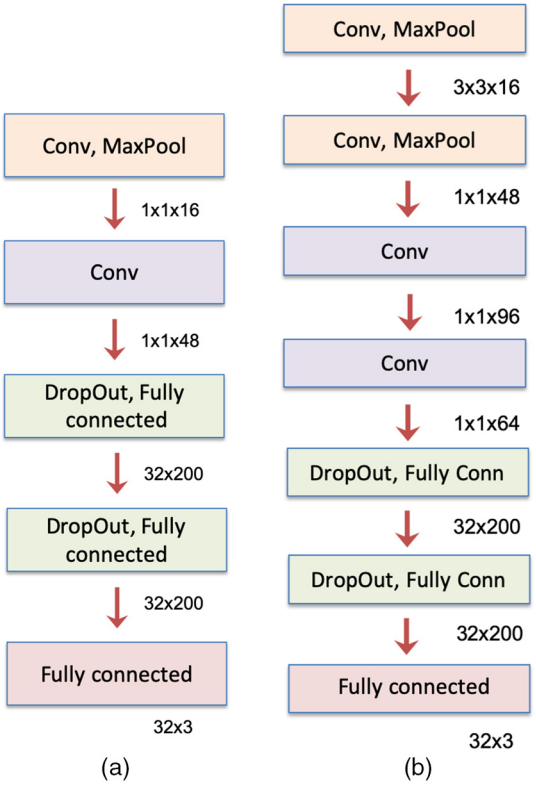


Fig. 2 DCNN architecture for the 8×8 and 4×4 patch resolutions (a) DCNN for 4×4 patch resolution and (b) DCNN for 8×8 patch resolution.

convolutional layers. Specifically, our final Sentinel-2 image (Yangon city, Myanmar) had a resolution of 10 m/pixel, thus an initial patch size of 32×32 pixels would imply information of 320×320 m. The initial 32×32 patch-size was chosen to resemble the 28×28 patch size of the DeepSat database to which the proposed architecture was first applied. However, as a patch of 320×320 m may contain information from a variety of scenes, decreasing the patch size would reveal unique scenes with a higher probability. However, decreasing the patch size required modification of the network architecture (as explained previously), thus, a rigorous analysis of patch sizes and network architectures was necessary.

In general, the proposed method differs from the state-of-the art method as it can be easily adapted to tackle a variety of problems with different image resolutions, by adding or removing layers. Thus, the proposed model performs well on the simple MNIST database, as well as on more complex databases, such as the very high DeepSat or the high EuroSat (both described in Sec. 3.2). Moreover, our proposal differs from the state-of-the art method since it trains the algorithm on a dataset with GT (EuroSat: images from Europe) and tests is on a completely unseen country of the same satellite (Myanmar), highlighting the possibility of automatically generating land-cover maps without the need of manually selecting GT pixels from the same image to train an algorithm.

3.2 *Datasets*

The DeepSat includes the SAT-6 and SAT-4 airborne datasets. In this study, we used the SAT-6 dataset that consists of 405,000 images patches of size 28×28 and six land-cover classes, namely barren land, trees, grassland, roads, buildings, and water bodies. The images consist of four bands, namely red, green, blue, and NIR.

The DeepSat database contains images of very high resolution (1 to 6 m per pixel), compared to the Sentinel-2 images (10 m per pixel). Thus, the first challenge was to explore whether the use of the DCNN model could be extended to successfully analyze (i.e., achieve accuracy higher than 80%) also other types of satellite images (e.g., Sentinel-2 images).

Hence, the DCNN model was also trained and tested in the newly published EuroSat database,[35] which consisted of a subset of Sentinel-2 satellite images with their truth GT. The GT covered ten classes, namely industrial, residential, annual crop, permanent crop, river, sea and lake, herbaceous vegetation, highway, pasture, and forest. The satellite images included patches of cities from 30 European countries. We performed object-based classification, i.e., we segmented the images into patches, as in Refs. 35 and 41. Although the patch size in Ref. 35 was of 64×64 pixels, we used 32×32, 16×16, 8×8, and 4×4 size patches to get the result into a finer resolution.

Often in the industrial sector we are called to segment new large land images from various countries in a strict timeline for our users. Thus, it is crucial to have a trained model on a dataset similar to the images we are called on to segment and directly use it for segmenting new images from other countries. Hence, in this work, once a good classification performance is achieved with the EuroSat (see Sec. 4), we train the model using all data from the EuroSat database and test it in a Sentinel-2 image of Yangon city and surroundings in Myanmar that we had previously classified at pixel level, in a semiautomatic way through region-based segmentation (see Sec. 3). The rationale is to investigate how the model performs in an unseen image of another continent and of the same satellite. Our reference data (RD) contain nine classes, namely urban fabric, main and country roads, high dense green area—forest/trees areas, medium dense green area—trees/agricultural areas, low dense green area—open/urban green areas, irrigated field areas, semi-irrigated field areas, open space with little or no vegetation, and shallow water. As a first step, we manually group the classes of both the EuroSat database and the Yangon city image and map them into three final classes, namely urban, nonurban, and water. Our classification of the Yangon city serves as RD for this image. We provide the final accuracy at the pixel level.

For both datasets, the images were not atmospherically corrected to explore the performance using the raw data, which in operational conditions would save time from the processing chain. However, all training images were standardized subtracting the mean and dividing by the standard deviation pixel-wise. The same procedure was applied also in the test data, using the mean and standard deviation obtained from the training data. For all datasets, four bands were used, namely RGB and NIR, similar to the initial DeepSat database.

3.3 *Performance Evaluation*

The classification results are evaluated in terms of the accuracy and the mean accuracy across classes. The accuracy refers to the correctly classified number of pixels with respect to the total number of pixels, whereas the mean accuracy across classes refers to the average accuracy estimated on each class. We will refer to the mean accuracy across classes simply as the mean accuracy for simplicity.

The problem that we often face in the industrial sector is the obvious lack of GT for the images we are called to segment, as we are asked to provide this information in a fast, automatic way. To evaluate the final segmentation result, an expert is manually labeling a set of randomly selected pixels. The UA and PA are then provided for these manually annotated pixels. This analysis is presented in Sec. 4.7.

To visualize the outcome, we output both the crisp class labels and the final probabilities. In the latter, we map the probability of each of the three classes into one color channel value for each patch. This type of representation allows the visualization of the possible uncertainties in the final classification map. If, for instance, a pixel presents a probability membership vector [0.5, 0.4, 0.1], where, e.g., 0.5 is the probability of the pixel belonging to the first class, that pixel is represented through RGB color values [127, 102, 25]. This RGB tuple would better represent the associated decision uncertainty than the tuple [255, 0, 0] that corresponds to the crisp presentation. The proposed visualization is valid for three classes, but different false composite images can be generated using different combinations of bands, as it is common practice when representing multispectral imagery or in radar polarimetry, each of them shedding different information and providing different visual insights. Each band might correspond to a different class probability score or to a combination of them. There is no particular constraint to the combinations one can generate to highlight relevant features of the scene.

3.4 *Applying a Threshold to Controversial Patches*

As often happens in machine learning problems, some data are close to classification boundaries and consequently are misclassified. In our case we refer to such data as controversial and in this study we wish to check the classification accuracy after removing a number of controversial data. Specifically, in our case, controversial patches are considered the ones for which the posterior probabilities of the DCNN are similar—the DCNN assigned a class to them, but with low confidence. The rationale of removing controversial patches is to find a compromise between accuracy and number of patches included in the analysis so that in a future stage we treat the controversial patches differently, e.g., by interpolating the classification results of neighboring patches, applying smoothing techniques, etc. In the case of many controversial patches, our proposed scheme could be applied as an initial mask to classify the noncontroversial patches and then apply other methods for the remaining controversial ones. The proposed analysis on the controversial patches also allows us to further investigate which parts of the image are misclassified and can shed more light into the reason of the misclassification.

To achieve this, we applied a threshold to the output probabilities of the final softmax layer. In particular, we sorted the probabilities per pixel for the three classes under consideration, took the difference between the two largest probabilities and applied a threshold to this difference. This threshold value represented the "accepted" distance between two posterior probabilities. If the difference (distance) between the highest probabilities was lower than the threshold, the pixel was considered controversial and set to color black, otherwise it was considered noncontroversial and was included in the accuracy estimation. This procedure is indicated in the following example. Take, for instance, a pixel with output probabilities [0.1, 0.3, 0.6] and a threshold th = 0.2, this pixel will be treated as a noncontroversial pixel, as the difference between its highest two probabilities (0.6 and 0.3) is 0.3, which is not smaller than the threshold.

Once all controversial pixels had been removed, we then calculated the pixel-wise accuracies of the remaining pixels. A higher difference and thus threshold indicated larger distance between the probabilities as "accepted" distance.

Choosing a proper threshold depends on the user's requirements and application. We anyhow suggest deriving the accuracy for a variety of thresholds and choose the one that better fits to the current needs. This is further discussed in Sec. 4.5.

4 Results

4.1 *Modified National Institute of Standards and Technology Database*

The MNIST data were downloaded from the TensorFlow example datasets.[43] The training data consisted of 55,000 grayscale (one channel) 28×28 images of handwritten digits. The test data consisted of 10,000 similar grayscale images of handwritten digits. The DCNN algorithm was initially applied to the MNIST dataset for validation purposes. The 10-class classification result on the test set yielded 99% accuracy, indicating that the DCNN was constructed successfully. Although there is no indication that a good performance on MNIST would lead to a good performance in land-cover classification, the reason that we first applied the algorithm on the MNIST dataset was for a "sanity" check of the algorithm itself, in order to make sure that the algorithm worked well for a very simple and controlled dataset before applying it to a more complex problem.

4.2 *DeepSat Database*

Regarding the DeepSat database, we trained the DCNN model on 234,000 images of 28×28 patches and tested it on 70,000 images. The final classification accuracy was 93.2%, verifying again that the model has been well implemented.

4.3 *EuroSat Database*

The next step was to apply the model with the newly developed EuroSat dataset (Sentinel-2 images dataset). In this case, we used 354,600 images of 32×32 patches for training and

10,000 images for testing. To minimize the impact on performance estimation from training set variability, we randomly selected the training and test sets and repeated this procedure 10 times. The average classification accuracy and its standard deviation from our first run are presented in Table 1, for each class and as a total.

We run the algorithm 10 times to get better statistics. The final mean accuracy and corresponding standard deviation achieved is 88.2 ± 0.5. One may notice from Table 1 that the average classification accuracy is high (>80%), which implies that the DCNN framework implemented for the DeepSat database with high resolution can be also applied to the lower resolution EuroSat database (Sentinel-2 data). The performance drops a bit, but the accuracy is still very high.

Another observation from Table 1 is that the highway class is often misclassified, and mainly confused with herbaceous vegetation and industrial, as observed from the confusion matrix presented in Fig. 3.

Table 1 Classification results from the EuroSat database with DCNN.

Class name	Average accuracy (%)	Standard deviation
Annual crop	91.3	2.2
Forest	98.6	0.8
Herbaceous	91.8	2.1
Highway	61.5	3.1
Industrial	84.4	5.5
Pasture	90.8	1.3
Permanent crop	85.8	4.9
Residential	93.2	1.2
River	80.8	1.5
Sea and lake	97.4	0.3
Average accuracy	**87.6**	**2.3**

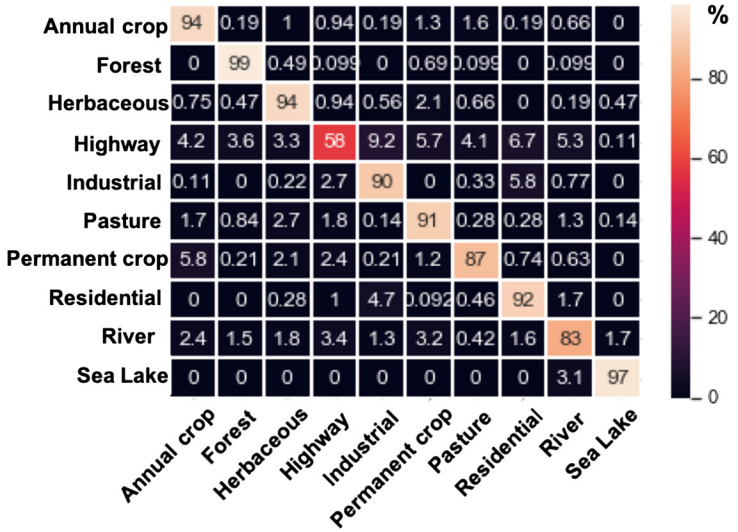
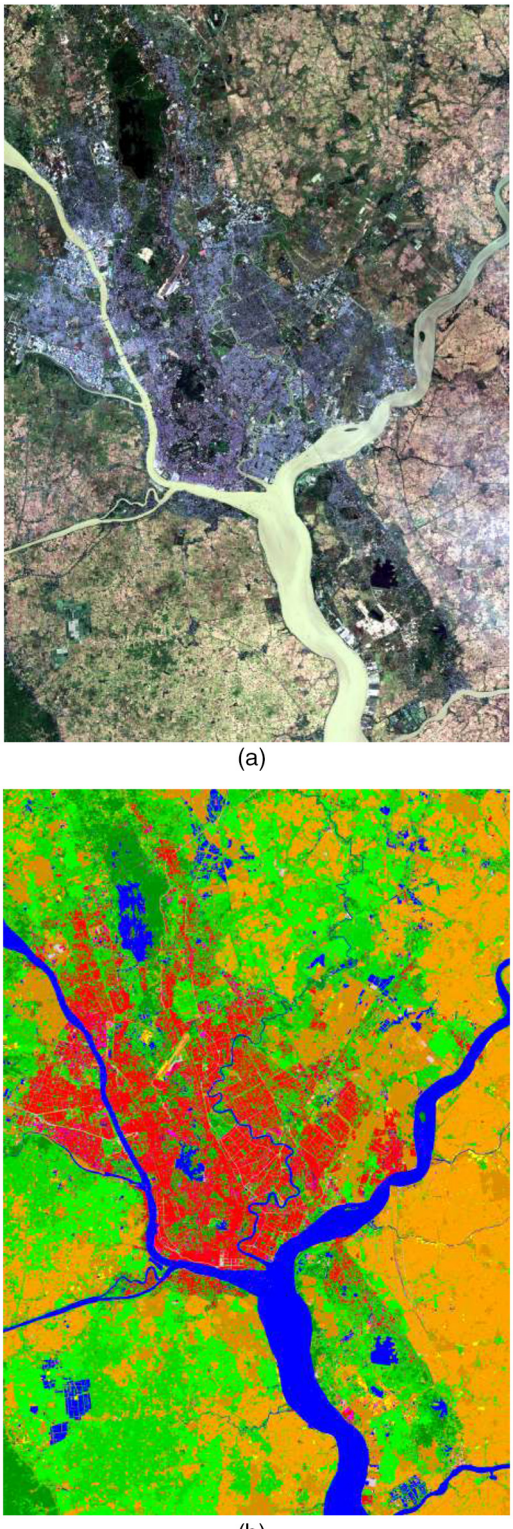


Fig. 3 Confusion matrix for the EuroSat dataset.

4.4 *Training on EuroSat Database and Testing in Myanmar*

As advanced above, our final test is to train the model using the EuroSat images and test it using an image from the Yangon city that we have previously classified and is not included in the training set. The image is presented in Fig. 4(a) and the RD with the nine classes are presented



(a)

(b)

Fig. 4 (a) Original image and (b) GT image, Yangon city, Myanmar.

Table 2 UA and PA of the RD with respect to GT annotated data.

Class name	UA (%)	PA (%)
Nonurban	99.1	71.4
Urban	70.6	97.6
Water	86.4	98.1

in Fig. 4(b). The RD was generated following Refs. 44 and 45. Specifically, a representative training set for each class has been selected supported by very high resolution imagery. Then, a set of pixel-based supervised classification methods (maximum likelihood, minimum distance, Mahalanobis distance, and parallelepiped classification) are applied to the test image and their accuracy is evaluated over a manually labeled statistical sample. The final classification image is produced by combining the output of the different classifiers based on their individual class accuracies. The classification product is further refined by filtering outliers through morphological operators and exploiting texture information to better separate urban areas.

Since the EuroSat dataset and the Myanmar image had labels with different classes, we have manually grouped the classes of each to fall into nonurban, water, and urban classes. The categorization is as follows: (1) nonurban: annual crop, pasture, permanent crop, herbaceous vegetation, and forest; (2) urban: highway and industrial; and (3) water: river, sea, and lake.

To verify the quality of the final three-class RD, an expert manually annotated around 400 randomly selected pixels for each class. The overall accuracy, as well as the UA and PA provided by the kappa coefficient were estimated between the RD and the annotated pixels. The overall accuracy of the RD was 85.5%, indicating that the RD had been acceptably generated. The UA and PA are presented in Table 2. The main source of error had been in the urban and nonurban classes with an error of 30% in the urban UA and in the nonurban PA.

4.4.1 *Results for 16 × 16, 8 × 8, and 4 × 4 patches*

In this case, each initial 64×64 patch is split into 16×16 patches with the same class label. The final classification accuracy results between the RD and the DCNN classified data are presented in Table 3 (second column). The accuracy is different from the mean accuracy due to the use of an unbalanced test set (the number of examples from each class differs).

The confusion matrix for this case is presented in Fig. 5. It is noticeable that the nonurban class is misclassified as urban, but as water as well. The classes water and urban are very well classified (>90% accuracy).

In the 8×8 patches case, each initial 64×64 patch is split into 8×8 patches with the same class label. The final classification accuracy results for this case are presented in Table 3. We notice from this table that although we have split the patches into smaller ones, the classification results are still in the similar range. Thus, we have already gained much in resolution and have not lost in accuracy, in part from the availability of 64 times more data.

Table 3 Classification results training the DCNN on EuroSat and testing it on Yangon city image, 16×16, 8×8, and 4×4 patches.

Class name/accuracy (%)	16×16 patches	8×8 patches	4×4 patches
Nonurban	66.8	66.4	74.6
Urban	96.2	94.6	95.4
Water	94.6	94.6	91.2
Accuracy	73.7	72.6	78.7
Mean accuracy across classes	85.9	85.2	87.1

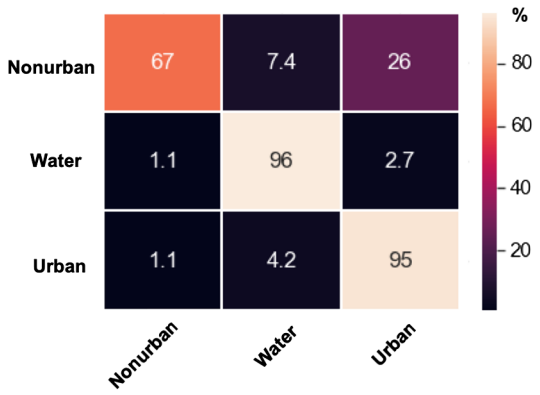


Fig. 5 Confusion matrix for the 16×16 patch case.

Similar to the previous cases, in the 4×4 patch case, each initial 64×64 patch is split into 4×4 patches with the same class label. The final classification accuracy results for this case are presented in Table 3 (fourth column).

Even though we decrease the patch size, the accuracy remains similar, while we gain significantly in spatial resolution. The output classification results as represented by the crisp output class for each of the three cases (i.e., RD, 16×16, 8×8, and 4×4 patch sizes) are presented in Fig. 6. The corresponding output probabilities, as described in Sec. 3.2, are presented in Fig. 7. The red color corresponds to the urban, the green to the nonurban, and the blue to the water class. One may notice from Figs. 6 and 7 that the 4×4 resolution gives a good approximation to the RD. Comparing Figs. 6 and 7, one may notice that the resolution seems more representative in Fig. 7, as there are finer details due to the assignation of probability values rather than raw class values.

4.5 *Controversial Patches*

We next apply a threshold to the posterior probabilities to remove the points considered as controversial, as described in Sec. 3.3. Please note that the threshold is not meant to quantify the misclassification but rather to mask out unreliable patches. In this sense, patches are generally tagged as unreliable using this approach when they correspond to a boundary between classes, or include pixels belonging to different classes. We can modify the threshold to further mask the output if we consider it still contains a large number of misclassified patches, at the cost of lower area coverage. This still can be useful as an additional layer of information to subsequent classification stages. The results are presented for the 4×4 patch case, as this case has delivered the best accuracy and resolution. Various thresholds are selected leading to a different number of points excluded from the final accuracy estimation. The results are displayed in Table 4. Two example images are presented in Fig. 8 for thresholds 0.3 and 0.7.

As seen from Table 4, although the classification accuracy increases with the increase of the threshold and the controversial pixels, for a threshold equal to 0.9, the classification accuracy of the nonurban class drops to 56%. We found out that the reason for this decrease is that there are some nonurban patches that are classified either as urban or as water with very high confidence. As many pixels have been removed for a threshold of 0.9 (86%), the misclassified ones with very high probability start to affect the overall accuracy of the nonurban class.

To further investigate the origin of misclassified pixels, we zoomed in some areas and present the original image, the RD of this area, and our output result in Fig. 9.

As seen in Fig. 9, the part that has been misclassified [white part in Fig. 9(c)] seems to actually be correctly classified as water with the DCNN and incorrectly indicated as nonurban in the RD [Fig. 9(b)]. The reason is that this part is the same as the rest of the parts of the same image that are indicated as water in the RD.

The controversial patches, as expected, reveal an increase in the accuracies with increasing thresholds, and thus, controversial pixels. However, since in our problem 80% is considered a good accuracy, a threshold of 0.3 (13% controversial pixels) is suggested as a good compromise

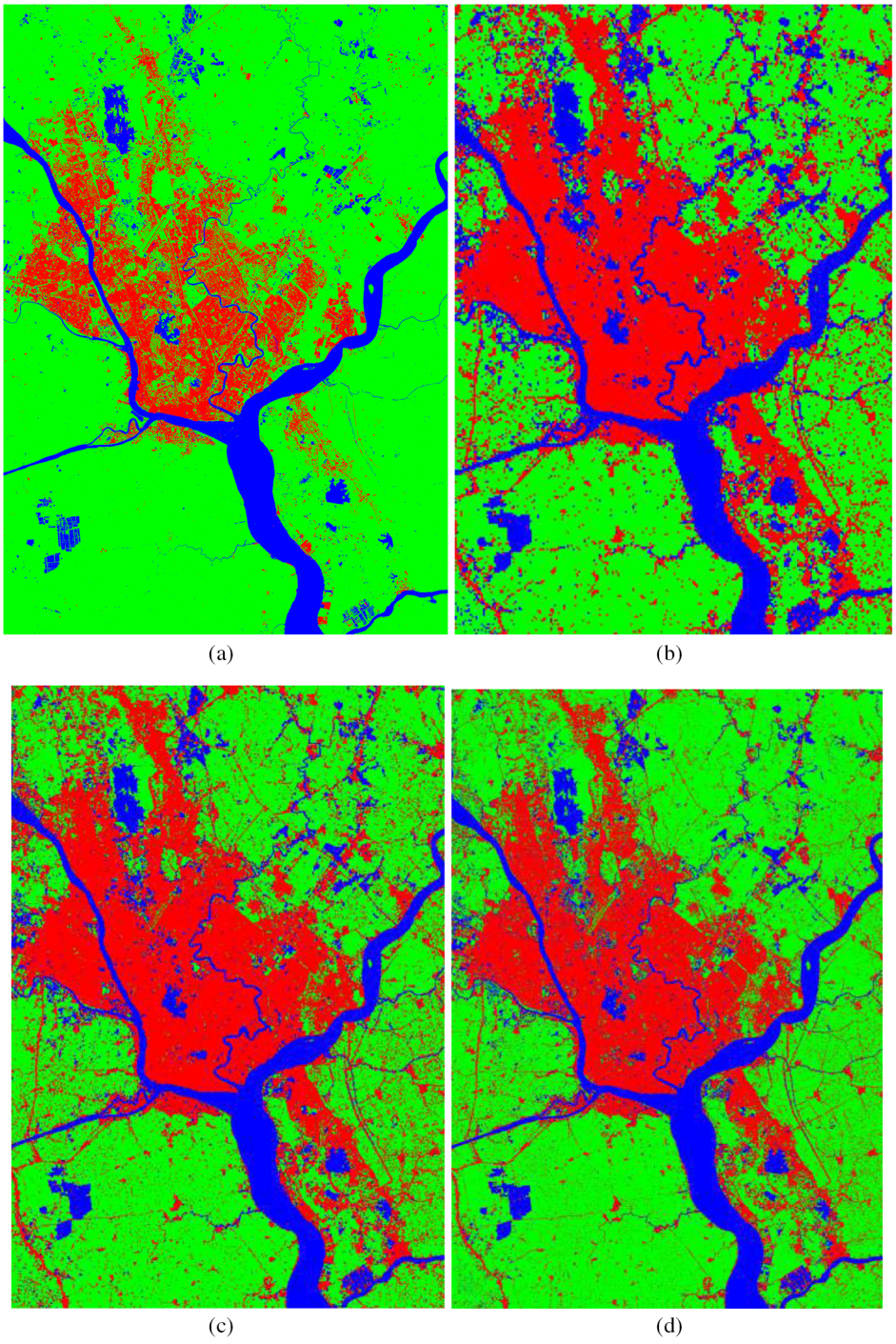


Fig. 6 Output classification results are represented by the crisp output class for RD and each of the three cases (i.e., 16×16, 8×8, and 4×4 patch sizes): (a) RD, (b) 16×16 patches, (c) 8×8 patches, and (d) 4×4 patches.

between accuracy and data loss for this problem, since the accuracy is higher than 80% for all classes. However, the threshold can be chosen depending on the users' needs. For instance, the proposed methodology could be used as an initial mask, i.e., assigning class labels only to pixels with high confidence, and apply other methods to the rest of the pixels, ranging from interpolating the classification results of neighboring pixels, applying smoothing techniques, etc. In this case, a user could choose to apply the proposed method to a smaller number of pixels, for which the probability that they belong to a certain class is very high.

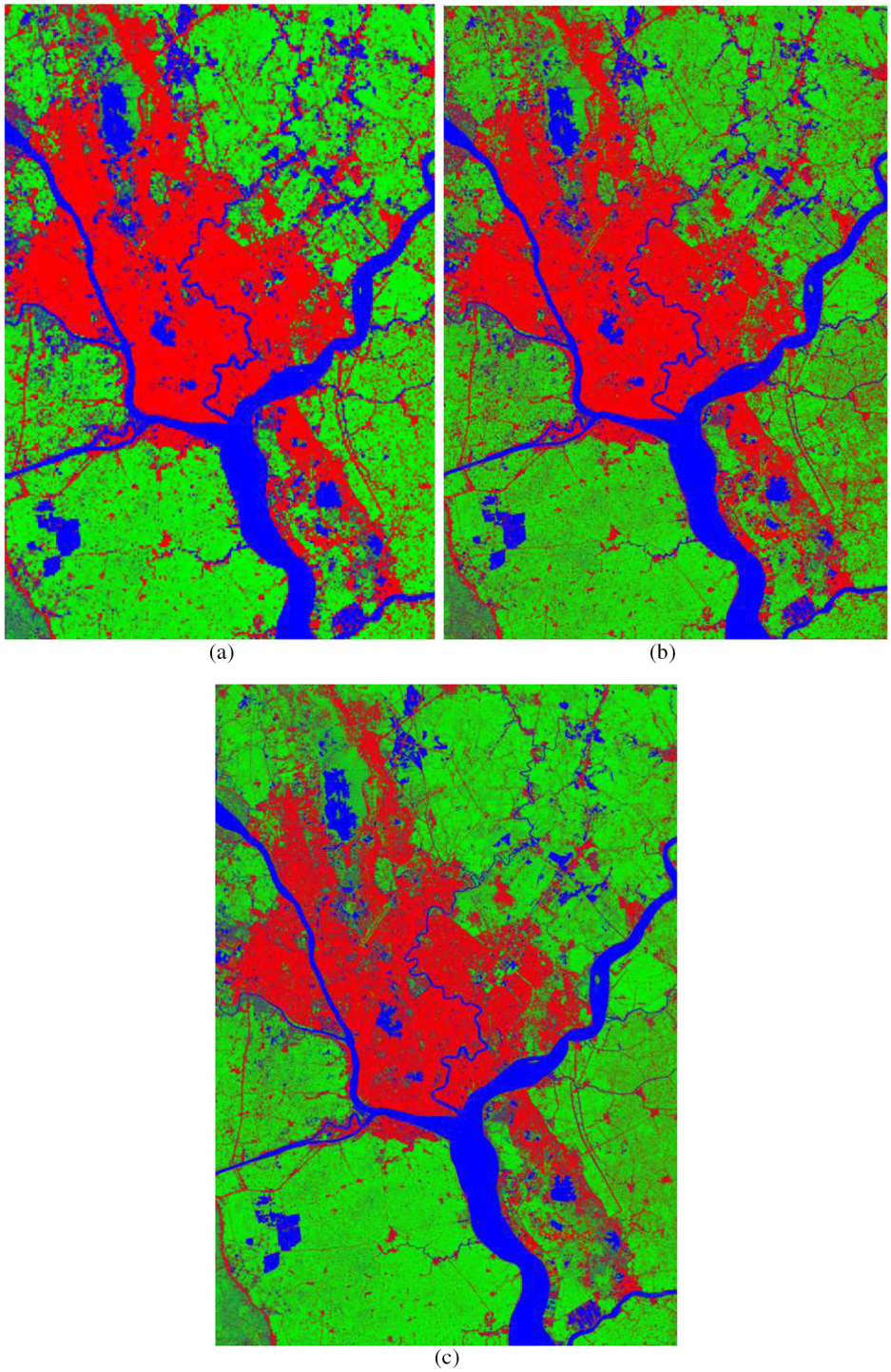


Fig. 7 Output classification results as represented by the output probabilities for each of the three cases (i.e., 16×16, 8×8, and 4×4 patch sizes): (a) 16×16 patches, (b) 8×8 patches, and (c) 4×4 patches.

Note than the automatic selection of an optimal threshold is out of the scope of this study, since it is pretty much linked to the application requirements. In any case, per-class ROC curves may be generated to evaluate sensitivity versus fall-out for different thresholds, and indicators such as the Youden's index could be considered to establish a trade-off between true- and false-positive rates. This is left for future work addressing different use case scenarios.

Table 4 Classification results for various thresholds, 4×4 patch case.

Threshold	Nonurban	Water	Urban	Accuracy	Mean accuracy	Percentage of controversial pixels (%)
0.05	75.8	95.6	91.9	79.8	87.7	2
0.1	76.9	95.8	92.3	80.7	88.3	5
0.2	78.9	96.2	94.2	82.4	89.3	9
0.3	80.2	96.6	93.9	83.8	91.1	13
0.4	81.7	97	94.6	85.2	91.1	18
0.5	82.8	97.3	95.4	86.4	91.9	24
0.6	83.2	97.7	96.4	87.4	92.4	32
0.7	83.2	98.2	97.4	87.9	92.9	42
0.8	83.3	98.8	98.3	89.5	93.5	58
0.9	56.4	99.5	98.5	92.1	84.8	86

4.6 *Visual Evaluation of the Results*

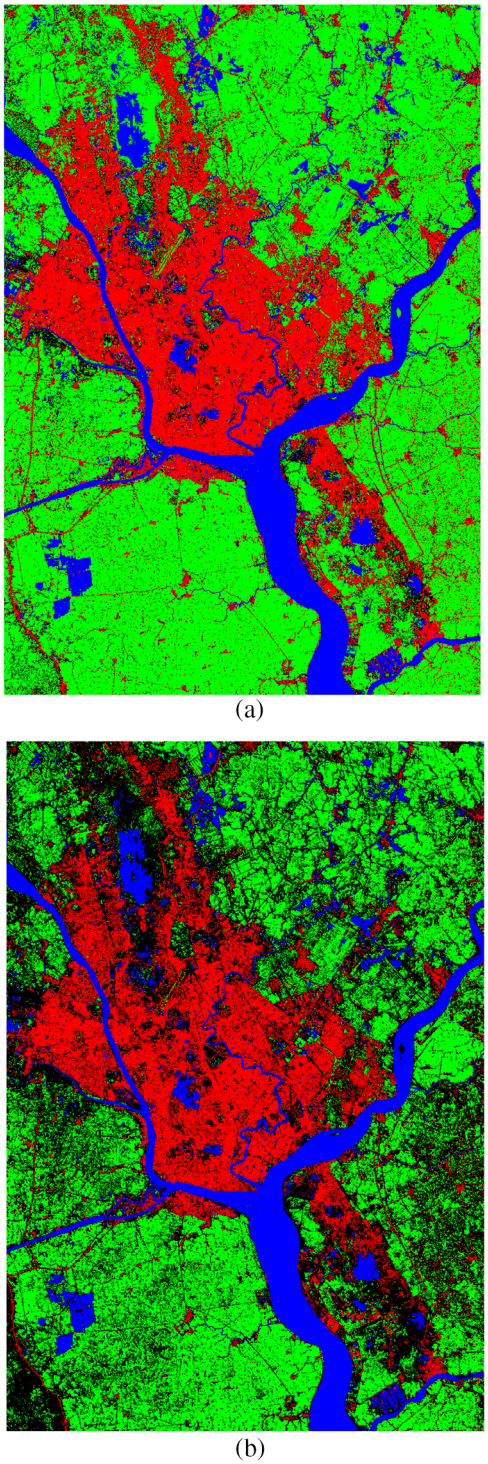
To further evaluate the DCNN performance and shed light into the parts for which either the algorithm did not perform well or the RD was not correctly indicated, we visualize some areas of the nonurban class since it is the class with the lowest classification accuracy. The images are presented in Figs. 10 and 11.

The examples in Figs. 10 and 11 present one case in which the DCNN failed and another one in which the DCNN performed better than the RD. In particular, as shown in Fig. 10, the trees that are along the road have not been captured by the DCNN, whereas they are correctly indicated as nonurban in the RD. On the contrary, in Fig. 11, the two crossed roads have been correctly captured by the DCNN, whereas they have been wrongly annotated as nonurban in the RD.

Although our DCNN seems to perform very well in identifying roads in the nonurban areas (e.g., Fig. 11), it does not perform well in identifying nonurban areas inside the city (e.g., Fig. 10). Our interpretation of this deficiency is the following: as the resolution of the image is around 10 m per pixel and the finest DCNN works with 4×4 patches, a patch inside the city contains more information of building, roads, etc. than nonurban areas, as happens for instance in Fig. 10(a). Since the DCNN infers the class label based also on neighboring pixels due to its convolutional properties, the final class label indicated does not capture the fine nonurban details. We believe that this algorithm would perform even better with higher-resolution images (e.g., 1 m/pixel), such as the DeepSat. However, there are cases in which these convolutional properties of the DCNN outperform the RD [e.g., Fig. 10(c)]. This applies to large homogeneous areas to which the DCNN has indicated the same label, whereas the RD has missed some parts.

4.7 *Comparison with Actual Ground Truth*

In the previous sections, it has been shown that the performance of the RD, although acceptable, is not perfect. To further evaluate the DCNN performance, we estimate the overall accuracy and UA and PA of the DCNN with respect to the manually annotated pixels. This reveals an accuracy of 84.3% and the UA and PA, as displayed in Table 5. The main source of error is identified in the urban PA and nonurban UA. However, although the performance is similar to the performance of the RD, the DCNN seems to perform better in terms of a compromise between the UA and the PA. In particular, the UA and PA of the RD are 97.6% and 70.62%, respectively (for the urban class), and 71.4% and 99%, respectively (for the nonurban class), whereas the corresponding values for the DCNN are 78.2% and 83.5%, respectively (for the urban class), and 84% and 79.8%, respectively (for the nonurban class).



(a)

(b)

Fig. 8 Images after removing the controversial patches (in black) for thresholds of 0.3 and 0.7. (a) Threshold of 0.3 and (b) threshold of 0.7.

Another important thing that needs to be pointed out is that the RD are generated based on the Myanmar image and is a time-consuming procedure. However, the DCNN classification outputs are based on training from a different dataset. Overall, these analyses point out the value of the DCNN models for satellite image region-based classification, both in terms of performance and in terms of speed since an already-trained model in a similar dataset can be used to automatically segment a new image with a good accuracy.

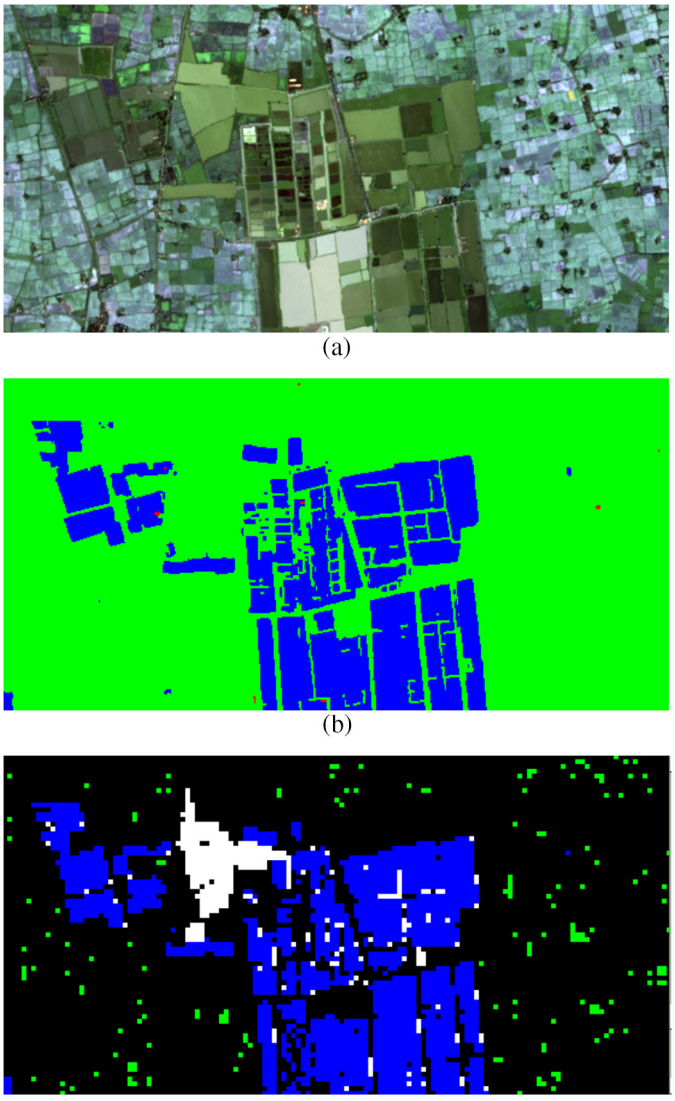


Fig. 9 Original, RD, and DCNN output (4×4 patches) for a nonurban area as shown in the RD. (a) Original, (b) RD, and (c) DCNN.

5 Discussion

Already from the introductory section one may notice that there are only a handful of studies using Sentinel-2 for land use land cover (LULC) mapping, and even less if we consider mixed rural and urban environments, which precludes a comprehensive comparative analysis between our method and other state-of-the-art methods. In general terms, supervised methods for LULC achieve high overall accuracies but they do so after careful tuning of the classifier parameters and after training them with GT samples extracted from the study area (at least 0.25% of the total area, as reported in Ref. 46). Unsupervised methods rarely perform as well as supervised models. Even if the dataset used for benchmarking (NWPU-RESISC45) is not entirely compatible with our study case, and neither are the target classes, the comprehensive review provided in Ref. 28 gives us an idea on the difference in performance between DL approaches and popular supervised methods. In particular, the latter report overall accuracy below 45%, whereas DL methods range between 70% and 80%, and exceed 80% after fine-tuning. Hence an 80% of overall accuracy seems a reasonable goal for the proposed method since we prioritize minimum user intervention versus accuracy. However, some limitations of this study that could be addressed in a future work in order to improve accuracy consist in the following: (1) we currently use a

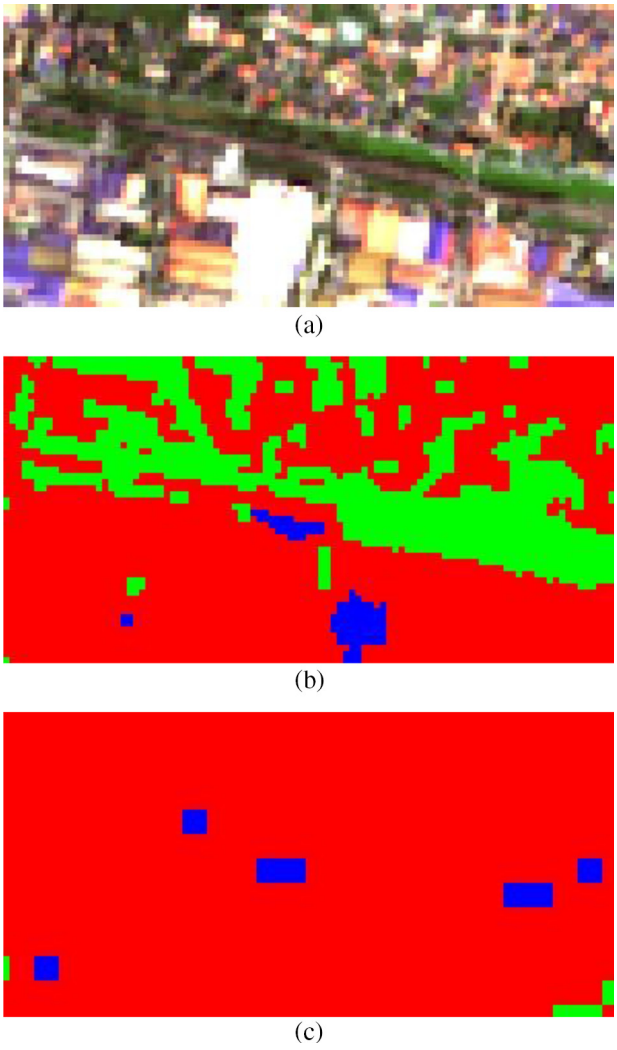


Fig. 10 Original, RD, and DCNN output (4×4 patches) for a nonurban area as shown in the RD. In this case the DCNN failed. (a) Original image, (b) RD, and (c) DCNN.

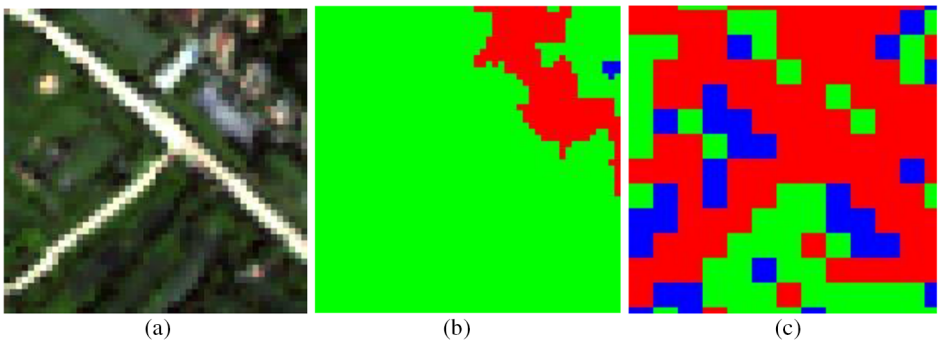


Fig. 11 Original, labeled RD, and DCNN output (4×4 patches) for a nonurban area as shown in the RD. In this case the DCNN performs better than the RD. (a) Original, (b) RD, and (c) DCNN.

prelabeled training set consisting 64×64 patches with a single label per patch; (2) we use different training area (Europe) than the test image (Myanmar); (3) we do not perform fine-tuning of classifier parameters nor do we apply further preprocessing to Sentinel-2 L1C data (e.g., no atmospheric correction); and (4) we are targeting a resolution actually finer than the training set patch size GT, so a drop of accuracy is to be expected. Thus, if we pretrain the

Table 5 UA and PA of the DCNN model with respect to GT annotated data.

Class name	UA (%)	PA (%)
Nonurban	79.8	84
Urban	83.5	78.2
Water	89.7	91

algorithm with a particular dataset and then use some samples of the target image to fine-tune it, then we expect the accuracy to increase, however, against absolute automation, as some manual effort would be needed to annotate pixels for the fine-tuning. In any case, we consider this study an exploratory work aiming to identify the advantages and limitations of such an approach, and providing ways on how we could overcome some of these limitations.

Further comparing the results with the corresponding EuroSat paper,[35] the overall accuracy achieved in that paper was 98.6% with 64×64 patch sizes, whereas in our case we achieved an accuracy of 87.6% with 32×32 patch sizes. Apart from the patch size, another difference between our implementation and the one proposed in the EuroSat paper was that the proposed method does not rely on any pretraining or fine-tuning of the network, as mentioned earlier, whereas the method proposed in Ref. 35 was pretrained on the ILSVRC-2012 dataset and later fine-tuned. However, the goal of this paper was to show that a DCNN in its simplest form could achieve high classification accuracy even though trained in one dataset and tested in another without any fine-tuning involved. Fine-tuning would, thus, be one of the suggested improvements of the proposed method. However, in the current paper we were interested in showing the value of DL methods for land cover with the minimum intervention possible.

In this study we also present a way of dealing with various patch sizes through removing intermediate layers of the network. As previously explained, the motivation for using various patch sizes is to achieve higher resolution outputs from a dataset with resolution of 10 m per pixel. Although the performance has not dropped by introducing finer patches and removing intermediate layers, it could be further improved using other strategies. For instance, in Refs. 30 and 47, the authors propose a fully convolutional network architecture and show that their network considers a large amount of context to provide fine classification maps. Other ways for providing the same output resolution as the original images include, for instance, removing the pooling layers from a standard CNN and adding dilated convolutions,[48] or using residual networks.[49,46] Although all these techniques look promising, an extended analysis should be carried out to identify the ones that could improve the accuracy for our problem.

To sum up, the main advantage of the patch-based segmentor model is that, by definition, it does not require a segmented training set. To date, it is still hard to find comprehensive, harmonized segmentation datasets for Sentinel-2 imagery, and they are costly to produce. Labeled patches are, on the other hand, much simpler to produce, but the larger the patches the less useful they are for applications such as LULC mapping. In this work we have explored the feasibility of using such datasets devised for patch-wise classification to provide a segmentation-like output, which of course requires getting to some compromise not only in terms of spatial resolution but also in terms of accuracy, hence the inclusion of the threshold approach to help identifying and masking out unreliable outputs.

6 Conclusions

We present a proof of concept on the use of DCNN for land classification. Using the properties of convolutions we have achieved high accuracies from training with a dataset of European cities' images and testing it on an image from Myanmar. Providing intermediate solutions to achieve higher accuracies through controversial pixels, we have presented an automated procedure for land-cover classification that performs well both in terms of accuracy and in terms of speed. Precisely, we have achieved a classification accuracy of 84.3% by training the model with an existing database and testing it to a completely new image, using a fully automated process

based on various DCNN architectures with the minimum possible interventions both in the images and in the DCNN models (e.g., pretraining/fine-tuning). We have compared the proposed DCNN architectures with some RD and have revealed that the five-layer architecture applied on 4×4 patches provides an approximation of 87.1% accuracy to the RD. Visual representation of the results in terms of the probability outputs for each class rather than raw class values provides more realistic image outputs with finer details. In case of need for further improving the accuracy with the cost of performing additional operations to some pixels (i.e., interpolation, smoothing techniques, etc. that would lead to a less automatic process), then removing 13% of the pixels using the proposed thresholding method to subject them to further operations leads to 91.1% of accuracy. The proposed thresholding method together with the proposed DCNN framework could be, thus, eventually integrated in a mapping software (e.g., geographic information system mapping software) to provide accurate maps in a few steps.

Our results and those of others indicate that DL is a powerful approach with an important future in the field. The combined availability of large datasets with emerging DL approaches and platforms that can exploit compositional features of data will no doubt find applications behind machine vision applications on land, such as in self-driving cars or web image data mining, where it is proving to be very immensely useful to applications using multichannel, multispectral space data for remote sensing of our planet or others.

References

1. L.-C. Chen et al., "DeepLab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFS," *IEEE Trans. Pattern Anal. Mach. Intell.* **40**(4), 834–848 (2018).
2. W. Zhang et al., "Deep convolutional neural networks for multi-modality isointense infant brain image segmentation," *NeuroImage* **108**, 214–224 (2015).
3. A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Adv. Neural Inf. Process, Syst.*, pp. 1097–1105 (2012).
4. G. Cheng, J. Han, and X. Lu, "Remote sensing image scene classification: benchmark and state of the art," *Proc. IEEE* **105**(10), 1865–1883 (2017).
5. B. Banerjee et al., "Unsupervised multi-spectral satellite image segmentation combining modified mean-shift and a new minimum spanning tree based clustering technique," *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **7**(3), 888–894 (2014).
6. J. Deng et al., "ImageNet: a large-scale hierarchical image database," in *IEEE Conf. Comput. Vision and Pattern Recognit. (CVPR)*, http://image-net.org/ (2009).
7. C. Szegedy et al., "Going deeper with convolutions," in *IEEE Conf. Comput. Vision and Pattern Recognit. (CVPR)*, pp. 1–9 (2015).
8. A. Albert, J. Kaur, and M. Gonzalez, "Using convolutional networks and satellite imagery to identify patterns in urban environments at a large scale," in *Proc. 23rd ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, pp. 1357–1366 (2017).
9. S. Basu et al., "DeepSat—a learning framework for satellite imagery," in *Proc. 23rd SIGSPATIAL Int. Conf. Adv. Geogr. Inf. Syst.* (2015).
10. M. Längkvist et al., "Classification and segmentation of satellite orthoimagery using convolutional neural networks," *Remote Sens* **8**(4), 329 (2016).
11. M. E. Paoletti et al., "A new deep convolutional neural network for fast hyperspectral image classification," *ISPRS J. Photogramm. Remote Sens.* **145**, 120–147 (2017).
12. J. E. Ball, D. T. Anderson, and C. S. Chan, "Comprehensive survey of deep learning in remote sensing: theories, tools, and challenges for the community," *J. Appl. Remote Sens.* **11**(4), 042609 (2017).
13. I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, MIT Press, Cambridge, Massachusetts (2016).
14. N. Kussul et al., "Deep learning classification of land cover and crop types using remote sensing data," *IEEE Geosci. Remote Sens. Lett.* **14**(5), 778–782 (2017).
15. A. Romero, C. Gatta, and G. Camps-Valls, "Unsupervised deep feature extraction for remote sensing image classification," *IEEE Trans. Geosci. Remote Sens.* **54**(3), 1349–1362 (2016).

16. W. Hu et al., "Deep convolutional neural networks for hyperspectral image classification," *J. Sens.* **2015**, 1–12 (2015).
17. Y. Chen et al., "Deep learning-based classification of hyperspectral data," *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **7**(6), 2094–2107 (2014).
18. Y. Chen, X. Zhao, and X. Jia, "Spectral–spatial classification of hyperspectral data based on deep belief network," *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **8**(6), 2381–2392 (2015).
19. C. Tao, H. Pan, Y. Li, and Z. Zou, "Unsupervised spectral–spatial feature learning with stacked sparse autoencoder for hyperspectral imagery classification," *IEEE Geosci. Remote Sens. Lett.* **12**(12), 2438–2442 (2015).
20. J. Geng et al., "High-resolution SAR image classification via deep convolutional autoencoders," *IEEE Geosci. Remote Sens. Lett.* **12**(11), 2351–2355 (2015).
21. K. Makantasis et al., "Deep supervised learning for hyperspectral data classification through convolutional neural networks," in *IEEE Int. Geosci. and Remote Sens. Symp. (IGARSS)*, pp. 4959–4962 (2015).
22. W. Zhao and S. Du, "Spectral–spatial feature extraction for hyperspectral image classification: a dimension reduction and deep learning approach," *IEEE Trans. Geosci. Remote Sens.* **54**(8), pp. 4544–4554 (2016).
23. L. Mou, P. Ghamisi, and X. X. Zhu, "Fully conv-deconv network for unsupervised spectral-spatial feature extraction of hyperspectral imagery via residual learning," in *IEEE Int. Geosci. and Remote Sens. Symp. (IGARSS)*, pp. 5181–5184 (2017).
24. L. Mou, P. Ghamisi, and X. X. Zhu, "Unsupervised spectral-spatial feature learning via deep residual conv-deconv network for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.* **56**(1), 391–406 (2018).
25. N. Audebert, B. Le Saux, and S. Lefevre, "How useful is region-based classification of remote sensing images in a deep learning framework?" in *IEEE Int. Geosci. and Remote Sens. Symp. (IGARSS)*, pp. 5091–5094 (2016).
26. E. Maggiori et al., "Fully convolutional neural networks for remote sensing image classification," in *IEEE Int. Geosci. and Remote Sens. Symp. (IGARSS)*, pp. 5071–5074 (2016).
27. D. Marmanis et al., "Deep learning Earth observation classification using ImageNet pretrained networks," *IEEE Geosci. Remote Sens. Lett.* **13**(1), 105–109 (2016).
28. X. X. Zhu et al., "Deep learning in remote sensing: a comprehensive review and list of resources," *IEEE Geosci. Remote Sens. Mag.* **5**(4), 8–36 (2017).
29. L. Zhang, L. Zhang, and B. Du, "Deep learning for remote sensing data: a technical tutorial on the state of the art," *IEEE Geosci. Remote Sens. Mag.* **4**(2), 22–40 (2016).
30. E. Maggiori et al., "Can semantic labeling methods generalize to any city? The Inria Aerial Image Labeling benchmark," in *IEEE Int. Symp. Geosci. and Remote Sens. (IGARSS)* (2017).
31. R. M. Anwer et al., "Binary patterns encoded convolutional neural networks for texture recognition and remote sensing scene classification," *ISPRS J. Photogramm. Remote Sens.* **138**, 74–85 (2018).
32. Q. Zou et al., "Deep learning based feature selection for remote sensing scene classification," *IEEE Geosci. Remote Sens. Lett.* **12**(11), 2321–2325 (2015).
33. Y. Yang and S. Newsam, "Bag-of-visual-words and spatial extensions for land-use classification," in *Proc. 18th SIGSPATIAL Int. Conf. Adv. Geogr. Inf. Syst.*, pp. 270–279 (2010).
34. G.-S. Xia et al., "AID: a benchmark data set for performance evaluation of aerial scene classification," *IEEE Trans. Geosci. Remote Sens.* **55**(7), 3965–3981 (2017).
35. P. Helber et al., "Introducing EuroSAT: a novel dataset and deep learning benchmark for land use and land cover classification," in *IEEE Int. Geoscience and Remote Sensing Symposium*, IEEE, Valencia, Spain, pp. 204–207 (2017).
36. P. Thanh Noi and M. Kappas, "Comparison of random forest, k-nearest neighbor, and support vector machine classifiers for land cover classification using Sentinel-2 imagery," *Sensors* **18**(1), 18 (2017).

37. B. Yao et al., "Nonlinear features of surface EEG showing systematic brain signal adaptations with muscle force and fatigue," *Brain Res.* **1272**, 89–98 (2009).
38. B. Liu et al., "A semi-supervised convolutional neural network for hyperspectral image classification," *Remote Sens. Lett.* **8**(9), 839–848 (2017).
39. G. Camps-Valls et al., "Advances in hyperspectral image classification: Earth monitoring with statistical learning methods," *IEEE Signal Process. Mag.* **31**(1), 45–54 (2014).
40. F. Ratle, G. Camps-Valls, and J. Weston, "Semisupervised neural networks for efficient hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.* **48**(5), 2271–2282 (2010).
41. M. Papadomanolaki et al., "Benchmarking deep learning frameworks for classification of very high resolution satellite multispectral data," *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* **III-7**, 83–88 (2016).
42. L. Deng, "The MNIST database of handwritten digit images for machine learning research [best of the web]," *IEEE Signal Process. Mag.* **29**(6), 141–142 (2012).
43. Y. LeCun et al., "Gradient-based learning applied to document recognition," *Proc. IEEE* **86**(11), 2278–2324, (1998).
44. B. E. Alhaddad, B. Arellano Ramos, and J. Roca Cladera, "Urban detection, delimitation and morphology: comparative analysis of selective "megacities"," *Int. Arch. Photogramm., Remote Sens. and Spat. Inf. Sci.*, **XXXIX-B7**, pp. 381–386 (2012).
45. B. I. Alhaddad, M. C. Burns, and J. R. Cladera, "Texture analysis for correcting and detecting classification structures in urban land uses; 'Metropolitan area case study-Spain'," in *Urban Remote Sens. Joint Event*, pp. 1–6 (2007).
46. H. Zhao et al., "Pyramid scene parsing network," in *IEEE Conf. Comput. Vision and Pattern Recognit. (CVPR)*, pp. 2881–2890 (2017).
47. N. Audebert, B. Le Saux, and S. Lefèvre, "Beyond RGB: very high resolution urban remote sensing with multimodal deep networks," *ISPRS J. Photogramm. Remote Sens.* **140**, 20–32 (2018).
48. F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," in *Int. Conf. Learning Representations (ICLR)*, 2016.
49. K. He et al., "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vision and Pattern Recognit.*, pp. 770–778 (2016).

Eleni Kroupi received her PhD in electrical engineering from Swiss Federal Institute of Technology in Lausanne (EPFL), Lausanne, Switzerland (2014) and her MSc degree in electrical and computer engineering from Aristotle University of Thessaloniki (AUTH), Thessaloniki, Greece (2010). She has worked as a postdoctoral researcher in the Applied Signal Processing Group at EPFL and in the Department of Psychology in University of Fribourg, Switzerland. Since 2016, she has been working as a neuroscience research engineer and data analyst at Starlab, Barcelona.

Maria Kesa earned her bachelor's degree in biology with a minor in mathematics from Tallinn University and her master's in applied mathematics from Tallinn University of Technology. In 2017, she did an internship at Starlab SL.

Víctor Diego Navarro-Sánchez received his engineer's degree in telecommunication (Miguel Hernandez University, 2009), master's degree in information technology (University of Alicante, 2011), and PhD in computer science (University of Alicante, 2014). He joined StarLab in 2013 as a research engineer and project manager. Since 2017, he also acts as R&D manager for StarLab Space Division. His research interests include polarimetric and interferometric SAR and multispectral methods for urban and natural environments monitoring and classification.

Salman Saeed completed his doctoral and postdoctoral research at the University of Surrey, UK (2014). He is the recipient of the IEEE GRSS J-STARS Paper Prize Award (2013) and is also a Fulbright scholar at the Central Florida Remote Sensing Laboratory, USA (2008–2009), where he contributed to the validation and calibration plan for the microwave radiometer on NASA's Aquarius/SAC-D mission. Prior to joining Starlab, he worked as a postdoctoral research fellow at the University of Exeter, UK (2013–2017).

Camille Pelloquin graduated in 2010 in general engineering including DSP. He participated in the Young Graduate Trainee Program at ESA/ESTEC in 2011, within the Sentinel-3 team, improving ground segment processing for the optical instruments OLCI and SLSTR. In 2013, he worked with the multi-satellite team in the Space Oceanography Division of CLS. He joined Starlab Barcelona in 2014 as research engineer and project manager.

Bahaa Alhaddad, after an MSc in remote sensing from space, obtained his PhD in urban management and valuations in 2009. From 2004 to 2011, as a staff member at the UPC Centre for Land Policy and Valuations, he has been involved in various European collaborative projects. Since 2011, he has been working as business developer in UK and an urban remote sensing expert at Starlab Limited.

Laura Moreno received her BSc degree in telecommunications engineering in 2003. In 2005, she obtained a grant of the Spanish Ministry of Science and Technology to work in the European Space Agency ESA/ESTEC, where she joined the Sentinel-1 team to work on level-2 products algorithms. In 2007, she joined Starlab, Barcelona, and is currently the director of the Space Applications and Services Department.

Aureli Soria-Frisch received his BSc degree from the University Ramon Llull (1992), his MSc degree from the Polytechnic University of Catalonia (1995), and his PhD from the Technical University Berlin. He is the director in the Neuroscience Department of Starlab. He was a project manager of the FP7 HIVE project, and PI of the MJFF grant for the development of Machine Learning PD biomarker discovery. He is the coordinator of the H2020 FET open project Luminous.

Giulio Ruffini graduated (math/physics) from UC Berkeley and obtained a PhD in physics from UC Davis (1995). In 2000, he cofounded Starlab to transform research into technologies with positive impact in the space and neuroscience sectors. Through several ESA projects, he and his team contributed to the development of emerging radar technologies, such as GNSS-R, SAR altimetry, or SAR interferometry. In 2011, he cofounded Neuroelectrics to deliver clinical EEG-tCS systems. He collaborates with teams worldwide developing applications of EEG-tCS, as in the Luminous FET EU project, studying consciousness.