POLITECNICO DI TORINO
Repository ISTITUZIONALE

# Improvement in Land Cover and Crop Classification based on Temporal Features Learning from Sentinel-2 Data Using Recurrent-Convolutional Neural Network (R-CNN)

*Original*
Improvement in Land Cover and Crop Classification based on Temporal Features Learning from Sentinel-2 Data Using Recurrent-Convolutional Neural Network (R-CNN) / Mazzia, Vittorio; Khaliq, Aleem; Chiaberge, Marcello. - In: APPLIED SCIENCES. - ISSN 2076-3417. - ELETTRONICO. - 10:1(2019), pp. 238-260. [10.3390/app10010238]

*Availability:*
This version is available at: 11583/2776772 since: 2020-01-09T15:11:13Z

*Publisher:*
MDPI

*Published*
DOI:10.3390/app10010238

*Terms of use:*

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

*Publisher copyright*

(Article begins on next page)

12 October 2024

applied sciences

MDPI

Article

# Improvement in Land Cover and Crop Classification based on Temporal Features Learning from Sentinel-2 Data Using Recurrent-Convolutional Neural Network (R-CNN)

**Vittorio Mazzia [1,2,3], Aleem Khaliq [1,2,*] and Marcello Chiaberge [1,2]**

1 Department of Electronics and Telecommunications, Politecnico di Torino, 10124 Turin, Italy; vittorio.mazzia@polito.it (V.M.); marcello.chiaberge@polito.it (M.C.)
2 PIC4SeR, Politecnico di Torino Interdepartmental Centre for Service Robotics, 10129 Turin, Italy
3 SmartData@PoliTo, Big Data and Data Science Laboratory, 10129 Turin, Italy
* Correspondence: aleem.khaliq@polito.it; Tel.: +39-389-041-9074

Received: 19 November 2019; Accepted: 23 December 2019; Published: 28 December 2019

**Abstract:** Understanding the use of current land cover, along with monitoring change over time, is vital for agronomists and agricultural agencies responsible for land management. The increasing spatial and temporal resolution of globally available satellite images, such as provided by Sentinel-2, creates new possibilities for researchers to use freely available multi-spectral optical images, with decametric spatial resolution and more frequent revisits for remote sensing applications such as land cover and crop classification (LC&CC), agricultural monitoring and management, environment monitoring. Existing solutions dedicated to cropland mapping can be categorized based on per-pixel based and object-based. However, it is still challenging when more classes of agricultural crops are considered at a massive scale. In this paper, a novel and optimal deep learning model for pixel-based LC&CC is developed and implemented based on Recurrent Neural Networks (RNN) in combination with Convolutional Neural Networks (CNN) using multi-temporal sentinel-2 imagery of central north part of Italy, which has diverse agricultural system dominated by economic crop types. The proposed methodology is capable of automated feature extraction by learning time correlation of multiple images, which reduces manual feature engineering and modeling crop phenological stages. Fifteen classes, including major agricultural crops, were considered in this study. We also tested other widely used traditional machine learning algorithms for comparison such as support vector machine SVM, random forest (RF), Kernal SVM, and gradient boosting machine, also called XGBoost. The overall accuracy achieved by our proposed Pixel R-CNN was 96.5%, which showed considerable improvements in comparison with existing mainstream methods. This study showed that Pixel R-CNN based model offers a highly accurate way to assess and employ time-series data for multi-temporal classification tasks.

**Keywords:** satellite imagery; deep Learning; pixel-based crops classification; recurrent neural networks; convolutional neural networks

## 1. Introduction

Significant increases in populations around the globe, increase demand in agricultural productivity, and, thus, precise land cover and crop classification and spatial distribution of various crops are becoming significant for governments, policymakers, and farmers to improve decision-making processes to manage agricultural practices and needs [1]. Crop maps are produced relatively at large scale, ranging from global [2], countrywide [3], and local level [4,5]. The growing need for agriculture in the management of sustainable natural resources becomes essential for the

development of effective cropland mapping and monitoring [6]. Group on Earth Observations (GEO), with its Integrated Global Observing Strategy (IGOS), also emphases on an operational system for monitoring global land covers and mapping spatial distribution of crops by using remote sensing imagery. Spatial information of the crop maps has been the main source for crop growth monitoring [7–9], water resources management [10], and decision making for policy makers to ensure food security [11].

Satellite and Geographic Information System (GIS) data have been an important source factor in establishing and improving the current systems that are responsible for developing and maintaining land cover and agricultural maps [12]. Freely available satellite data offers one of the most applied sources for mapping agricultural land and assessing important indices that describe conditions of crop fields [13]. Recently launched sentinel-2 is equipped with a multispectral imager that can provide up to 10 m per pixel spatial resolution with the revisit time of 5 days, which offers a great opportunity to be exploited in the remote sensing domain.

Multispectral time series data acquired from MODIS and LANDSAT have been widely used in many agricultural applications such as crop yield prediction [14], landcover and crop classification [15], leaf area index estimation [16], plant height estimation [17], vegetation variability assessment [18], and many more. Two different data sources can also be used together to extract more features that lead to improving results. For example, Landsat-8 and sentinel-1 used together for LC&CC [19].

There are some supervised or unsupervised algorithms for mapping cropland using mono or multi-temporal images [20,21]. Multi-temporal images have already proven to gain better performance than mono temporal mapping methods [22]. The imagery used for only key phenological stages s proved to be sufficient for crop area estimation [23,24]. It has also found in [25], that reducing time-series length affects the average accuracy of the classifier. Crop patterns were established using the Enhanced Vegetation Index derived from 250 meters MODIS-Terra time series data and used to classify some major crops like corn, cotton, and soybean in Brazil [26]. Centimetric resolution imagery is available at the cost of the high price of commercial satellite imagery or with the extensive UAV flight campaigns to cover large area during the whole crop cycle to get better spatial and temporal details. However, most of the studies used moderate spatial resolution (10–30 m) freely available satellite imagery for land cover mapping due to their high spectral and temporal resolution which is difficult in the case of UAV and high-resolution satellite imagery.

Other than multispectral time series data, several vegetation indices (VIs) derived from different spectral bands have been exploited and used to enrich the feature space for vegetation assessment and monitoring [27,28]. VIs such as the normalized difference vegetation index (NDVI), normalized difference water index (NDWI), enhanced vegetation indexes (EVI), textural features, such as grey level co-occurrence matrix (GLCM), statistical features, such as mean, standard deviation, inertial moment are the features more frequently used for crop classification. It is possible to increase the accuracy of the algorithms also using ancillary data such as elevation, census data, road density, or coverage. Nevertheless, all these derived features, along with the phenological metrics involve a huge volume of data, which may increase computational complexity with little improvement in accuracy [29]. Several feature selection methods have been proposed [29] to deal with this problem. In [30], various features have been derived from the MODIS time series and best feature selection has been made using the random forest algorithm.

LC&CC can also be classified as pixel-based or object-based. Object-based image analysis (OBIA), described by Blaschke, that segmentation of satellite images into homogeneous image segments can be achieved with high-resolution sensors [31]. The various object-based classification has been proposed to produce crop maps using satellite imagery [32–34].

In this work, we proposed a unique deep neural network architecture for LC&CC, which comprises of Recurrent Neural Network (RNN) that extracts temporal correlations from time series of sentinel-2 data in combination with Convolutional Neural Network (CNN) that analyzes and encapsulate the crops pattern through its filters. The remainder of this paper is organized as follows.

Section 2 briefs about related work done for the LC&CC along with an overview of RNN and CNN. Section 3 provides an overview of the raw data collected and exploited during the research. Section 4 provides detailed information on the proposed model and the training strategies. Section 5 contains a complete description of the experiments, results, and discussion along with the comparison with previous state-of-the-art results. Finally, Section 6 draws some conclusions.

## 2. Related Work

### 2.1. Temporal Feature Representation

There are various studies proposed in the past to address LC&CC. A more common approach adopted for classification tasks is to extract temporal features and phenological metrics from the VIs time series derived from remotely sensed imagery. There are also some simple statistics and threshold-based procedures used to calculate vegetation related metrics such as Maximum VI and time of peak VI [35,36], which have improved classification accuracy when compared to using only VI as features [37]. More complex methods have been adapted to extract temporal features and patterns to address the vegetation phenology [38]. Further, the time series of VI represented by a set of functions [39], linear regression [40], Markov model [41], and curve-fitting functions. Sigmoid function has been exploited by [42,43] and achieved better results due to its robustness and ease to derive phenological features for the characterization of vegetation variability [44]. Although above-mentioned methods of temporal feature extraction offer many alternatives and flexibilities in deployment to assess vegetation dynamics, in practice, there are some important factors such as manually designed model and feature extraction, intra-class variability, uncertain atmospheric conditions, empirical seasonal patterns, which make the selection of such methods more difficult. Thus, an appropriate approach is needed to fully utilize the sequential information from time series of VI to extract temporal patterns. As our proposed DNN architecture is based on pixel classification, therefore the following subsections will provide relevant studies and description

### 2.2. Pixel-Based Crops Classification

A detailed review of the state-of-the-art supervised pixel-based methods for land cover mapping was performed in [45]. It was found that the support vector machine (SVM) for mono temporal image classification was the most efficient in terms of overall accuracy (OA) of about 75%. The second approach was the neural networks (NN) based classifier with almost the same OA 74%. SVM is complex and resource-consuming for time series multispectral data applications with broad area classification. Another common approach in remote sensing applications is the random forest (RF)-based classifiers [46]. Nevertheless, multiple features should be derived to feed the RF classifier for effective use. Deep Learning (DL) is a branch of machine learning, and it is a powerful tool that is being widely used in solving a wide range of problems related to signal processing, computer vision, image processing, image understanding, and natural language processing [47]. The main idea is to discover not only the mapping from representation to output but also the representation itself. That is achieved by breaking a complex problem into a series of simple mappings, each described by a different layer of the model, and then composing them in a hierarchical fashion. A large number of state of the art models, frameworks, architecture, and benchmark databases of reference imagery exist for image classification domain.

### 2.3. Recurrent Neural Network (RNN)

Sequence data analysis is an important aspect in many domains, ranging from natural language processing, handwriting recognition, image captioning, to robot automation. In recent years, Recurrent Neural Networks (RNNs) have proven to be a fundamental tool for sequence learning [48], allowing to represent information from the context window of hundreds of elements. Moreover, the research community has, over the years, come up with different techniques to overcome the difficulty of training

over many time steps. For example, long short-term memory (LSTM) [49] and gated recurrent unit (GRU) [50] based architectures have proven ground-breaking achievements [51,52], in comparison to standard RNN models. In remote sensing applications, RNNs are commonly used when sequential data analysis is needed. For example, Lyu et al. [53] employed RNN to use sequential properties such as spectral correlation and intra-bands variability of multispectral data. They further used the LSTM model to learn a combined spectral-temporal feature representation from an image pair acquired at two different dates for change detection [54].

### 2.4. Convolutional Neural Network (CNN)

Convolutional Neural Networks (CNNs) date back decades [55], emerging from the study of the brain's visual cortex [56] and classical concepts of computer vision theory [57,58]. Since the 1990s, these have been applied successfully in image classification [55]. However, due to technical constraints such as mainly lack of hardware performance, the large amount of data, and theoretical limitations, CNNs did not scale to large applications. Nevertheless, Geoffrey Hinton and his team demonstrated at the annual ImageNet ILSVRC [59] competition the feasibility to train large architectures capable of learning several layers of features with increasingly abstract internal representations [60]. Since that breakthrough achievement, CNNs became the ultimate symbol of the Deep Learning [47] revolution, incarnating all those concepts that underpin the entire novel movement.
In recent years, DL was widely used in data mining and remote sensing applications. In particular, image classification studies exploited several DL architectures due to their flexibility in feature representation, and automation capability for end-to-end learning. In DL models, features can be automatically extracted for classification tasks without feature crafting algorithms by integrating autoencoders [61,62]. 2D CNNs have been broadly used in remote sensing studies to extract spatial features from high-resolution images for object detection and image segmentation [63–65]. In crop classification, 2D convolution in the spatial domain performed better than 1D convolution in the spectral-domain [66]. These studies formed multiple convolutional layers to extract spatial and spectral features from remotely sensed imagery.

## 3. Study Area and Data

The study site near Carpi, Emilia-Romagna, situated in the center-north part of Italy with central coordinates 44°47′01″ N, 10°59′37″ E was considered for LC& CC shown in Figure 1. The Emilia-Romagna region is one of the most fertile plains of Italy. An area almost 2640 km$^2$ was considered, which covers diverse cropland. The major crop fields in this region are maize, lucerne, barley, wheat, and vineyards. The yearly averaged temperature and precipitation are 14 °C and 843 mm for this region. Most of the farmers practice single cropping in this area.

To know about the spatial distribution of crops, we deeply studied Land Use Cover Area frame statistical Survey (LUCAS) and extracted all the information we need for ground truth data. LUCAS was carried out by Eurostat to be able to monitor agriculture, climate change, biodiversity, forest, and water for almost all over Europe [67].

The technical reference document of LUCAS-2015 was used to prepare the ground truth data. Microdata that contains spatial information of crops and several land cover types along with the geo-coordinates for the considered region was imported in Quantum Geographic information system (QGIS) software, an Open-source software used for visualization, editing, analysis of geographical data. The selection of pixel was made manually by overlapping images and LUCAS data, so a proper amount of ground truth pixels were extracted for training and testing the algorithm. Several examples of ground truth fields parcels are illustrated in Figure 2. The sentinel-2 mission consists of twin polar-orbiting satellites launched by European Space Agency (ESA) in 2015 and can be used in various application areas such as land cover change detection, natural disaster monitoring, forest monitoring, and most importantly in agricultural monitoring and management [68].

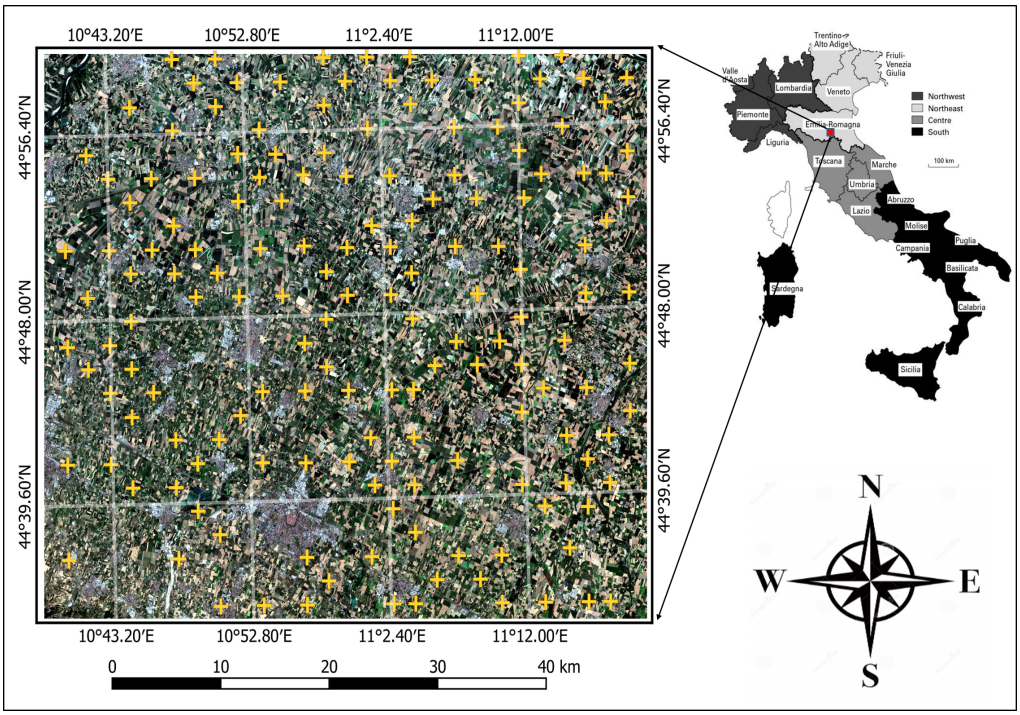


**Figure 1.** The study site is located in Carpi, region Emilia-Romagna is shown with the geo-coordinates (WGS84). RGB image composite derived from sentinel-2 imagery acquired in August-2015 is shown and the yellow marker showing geo-locations of ground truth land cover extracted from the Land Use and Coverage Area frame Survey (LUCAS-2015).

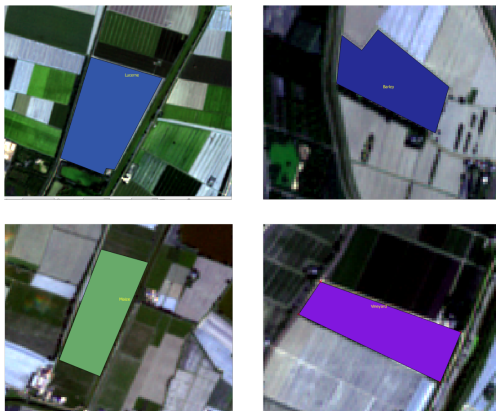

**Figure 2.** Several examples of zoomed in parts of crop classes considered as ground truth. Shape files are used to extract pixels for reference data.

It is equipped with multi-spectral optical sensors that capture 13 bands of different wavelengths. We used only high-resolution bands that have 10 m/pixel resolution shown in Table 1. It also has a high revisit time of ten days at the equator and five days with twin satellites (Sentinel-2A, Sentinel-2B). It became more popular in remote sensing community due to fact that it possesses various key features such as, free access to data products available at ESA Sentinel Scientific Data Hub with reasonable spatial resolution (which is 10 m for Red, Green, Blue, and Near Infrared bands), high revisit time and reasonable spectral resolution among other available free data sources. In our study, we used ten multitemporal sentinel-2 images reported in Table 2, which are well co-registered from July-2015 to July-2016 with close to zero cloud coverage. The initial image selection was performed based on the cloudy pixel contribution at the granule level. This pre-screening was followed by further visual inspection of scenes and resulted in a multi-temporal layer stack of ten images. Sentinel Application Platform (SNAP) v5.0 along with sen2core v 2.5.1 were used to apply radiometric and geometric

corrections to acquire Bottom of Atmosphere (BOA) Level 2A images from Top of Atmosphere (TOA) Level 1C. Further details about geometric, radiometric correction algorithms used in sen2cor can be found in [69]. Bands with 10 meters/pixel along with the derived Normalized Difference Vegetation Index (NDVI) were used for experiments, as shown in Table 1.

**Table 1.** Bands used in this study.

| Bands Used | Description | Central Wavelength (μm) | Resolution (m) |
|---|---|---|---|
| Band 2 | Blue | 0.49 | 10 |
| Band 3 | Green | 0.56 | 10 |
| Band 4 | Red | 0.665 | 10 |
| Band 8 | Near-infrared | 0.705 | 10 |
| NDVI | (Band8-Band4)/(Band8+Band4) | - | 10 |

**Table 2.** Sentinel-2 data acquisition.

| Date | Doy | Sensing Orbit # | Cloud Pixel Percentage |
|---|---|---|---|
| 7/4/2015 | 185 | 22-Descending | 0 |
| 8/3/2015 | 215 | 22-Descending | 0.384 |
| 9/2/2015 | 245 | 22-Descending | 4.795 |
| 9/12/2015 | 255 | 22-Descending | 7.397 |
| 10/22/2015 | 295 | 22-Descending | 7.606 |
| 2/19/2016 | 50 | 22-Descending | 5.8 |
| 3/20/2016 | 80 | 22-Descending | 19.866 |
| 4/29/2016 | 120 | 22-Descending | 18.61 |
| 6/18/2016 | 170 | 22-Descending | 15.52 |
| 7/18/2016 | 200 | 22-Descending | 0 |

## 4. Convolutional and Recurrent Neural Networks for Pixel-Based Crops Classification

### 4.1. Formulation

A single multi-temporal, multi-spectral pixel can be represented as a two-dimensional matrix $X^{(i)} \in \mathbb{R}^{t*b}$ where $t$ and $b$ are the number of time steps and spectral bands, respectively. Our goal is to compute from $X^{(i)}$ a probability distribution $F(X^{(i)})$ consisting of $K$ probabilities, where $K$ is equal to the number of classes. To achieve this objective, we propose a compact representation learning architecture composed of three main building blocks:

- **Time correlation representations**—this operation extracts temporal correlations from multi-spectral, temporal pixels $X^{(i)}$ exploiting a sequence-to-sequence recurrent neural network based on long short-term memory (LSTM) cells. A final time-distributed layer is used to compress and maintain a sequence like structure, preserving the multidimensionality nature of the data. In this way, it is possible to take advantage of temporal and spectral correlations simultaneously.
- **Temporal pattern extraction**—this operation consists of a series of convolutional operations followed by rectifier activation functions that non linearly maps each elaborated temporal and spectral patterns onto high dimensional representations. So, RNN output temporal sequences are processed by a subsequent cascade of filters, which in a hierarchical fashion, extracts essential features for the successive stage.
- **Multiclass classification**—this final operation maps the feature space with a probability distribution $F(X^{(i)})$ with $K$ different probabilities, where $K$, as previously stated, is equal to the number of classes.

The comprehensive pipeline of operations constitutes a lightweight, compact architecture able to non-linearly map multi-temporal information with its intrinsic nature, achieving results considerably better than previous state-of-the-art solutions. Human brain mental imagery studies [70], where images

are a form of internal neural representation, inspired the presented architecture. Moreover, the joint effort of RNN and CNN distributes the knowledge representation through the entire model, exploiting one of the most powerful characteristics of deep learning known as distributed learning. An overview of the overall model, dubbed Pixel R-CNN, is depicted in Figure 3. Each pixel is extracted contemporary from all images taken at different time steps $t$ with all its spectral bands $b$. In this way, it is possible to create an instance $X^{(i)}$, which can feed the first layer of the network. Firstly, the model extracts temporal representations from the input sample. Subsequently, these temporal features are further enriched by the convolutional layers that patterns in a hierarchical manner. The overall model act as a function $F(X^{(i)})$ that map the input sample with its related probabilities $K$. So, evaluating the probability distribution is possible to identify the class belonging to the input sample.

It is worth noticing that this model is known as unrolled through time representation. Indeed, only after all time steps have been processed, CNN is able to analyze and transform the temporal pattern. In the following subsections, we are going to describe in detail each individual block.

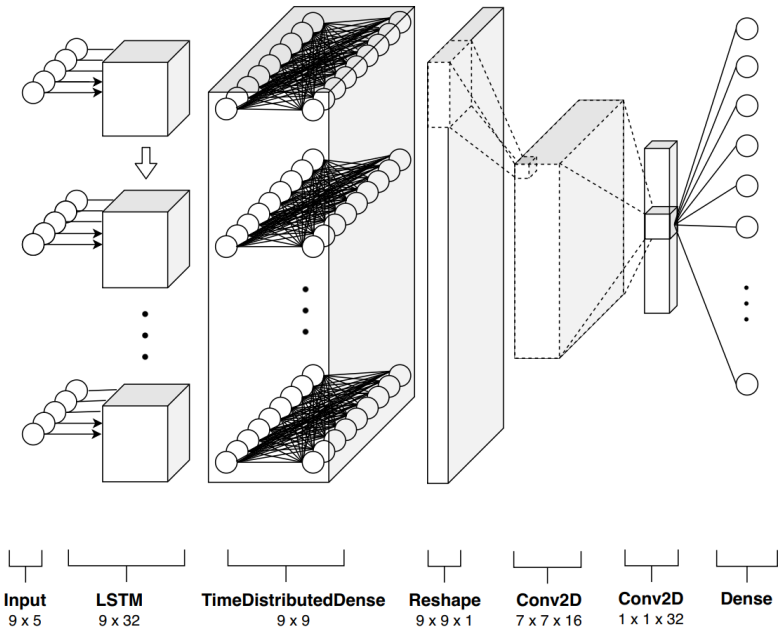


**Figure 3.** An overview of the pixel recurrent-convolutional neural networks (R-CNN) model used for classification. Given a multi-temporal, multi-spectral input pixel $X^{(i)}$, the first layer of long short-term memory (LSTM) units extracts sequences of temporal patterns. A stack of convolutional layers hierarchically processes the temporal information.

#### 4.1.1. Time Correlation Representation

Nowadays, a popular strategy in time series data analysis is the use of RNNs that have proven excellent results in many fields of the application over the years. Looking at the simplest possible RNN shown in Figure 4, composed of just one layer, it looks very similar to a feedforward neural network, except it also has a connection going backward. Indeed, the layer is not only fed by an input vector $x^{(i)}$, but it also receives $h^{(i)}$ (cell state), which is equal to the output neuron itself, $y^{(i)}$. So, at each time step $t$, this recurrent layer receives an input 1-D array $x_t^{(i)}$ as well as its own output from the previous time step, $y_{(t-1)}^{(i)}$. In general, since the output of a recurrent neuron at time step $t$ is a function of all inputs from previous time steps, it has, intuitively, a sort of memory that influences all successive outputs. In this example, it is straightforward to compute a cell's output, as shown in Equation (1).

$$y_t^{(i)} = \phi(x_t^{(i)} \cdot W_x + y_{(t-1)}^{(i)} \cdot W_y + b), \tag{1}$$

where, in the context of this research, $x_t^{(i)} \in \mathbb{R}^{(1*b)}$ is a single time step of a pixel with $n_{inputs}$ equal to the number of spectral bands b. $y_t^{(i)}$ and $y_{(t-1)}^{(i)}$ are the output of the layer at time $t$ and $t-1$, respectively, $W_x$ and $W_y$ are the weights matrices. It is important to point out that $y_t$ as $x_t^{(i)}$ are vectors and they can have an arbitrary number of elements, but the representation Figure 4 does not change. Simply, all neurons are hidden in the depth dimension. Unfortunately, the basic cell just described suffers from major limitations, but most of all it is the fact that, during training, the gradient of the loss function gradually fades away. For this reason, for the time correlation representation, we adopted a more elaborated cell known as the peephole LSTM unit, see Figure 5. That is an improved variation of the concept proposed in 1997 by Sepp Hochreiter and Jurgen Schmidhuber [49]. The key idea is that the network can learn what to store in a long-term state, $c_{(t)}$ what to throw away and what to use for the current state $h_{(t)}$ and $y_{(t)}$ that, as for the basic unit, are equal. That is performed with simple element-wise multiplications working as "*valves*" for the fluxes of information. Those elements, $V_1$, $V_2$, and $V_3$ are controlled by fully connected (FC) layers that have as input the current input state $x_{(t)}$ and the previous short-term memory term $h_{(t-1)}$. Moreover, for the peephole LSTM cell, the previous long-term state $c_{(t-1)}$ is added as an input to the FC of the forgot gate,$V_1$, and the input gate, $V_2$. Finally, the current long-term state $c_t$ is added as an input to the FC of the output gate. All "gates controllers" have sigmoid as activation functions (green boxes) instead of tanh ones to process the signals themselves (red boxes). So, to summarize, a peephole LSTM block has three signals as input and output; two are the standard input state $x_{(t)}$ and cell output $y_{(t)}$. Instead, $c$ and $h$ are the long and short-term state, respectively, that the unit, utilizing its internal controllers and valves, can feed with useful information. Formally, as for the basic cell seen before, Equations (2) and (7) summarizes how to compute the cell's long-term state, its short-term state, and its output at each time step for a single instance.

$$i_{(t)} = \sigma(W_{ci}^T \cdot c_{(t-1)} + W_{hi}^T \cdot h_{(t-1)} + W_{xi}^T \cdot x_{(t)}^{(i)} + b_i), \tag{2}$$

$$f_{(t)} = \sigma(W_{cf}^T \cdot c_{(t-1)} + W_{hf}^T \cdot h_{(t-1)} + W_{xf}^T \cdot x_{(t)}^{(i)} + b_f), \tag{3}$$

$$o_{(t)} = \sigma(W_{co}^T \cdot c_{(t)} + W_{ho}^T \cdot h_{(t-1)} + W_{xo}^T \cdot x_{(t)}^{(i)} + b_o), \tag{4}$$

$$g_{(t)} = \tanh(W_{hg}^T \cdot h_{(t-1)} + W_{xg}^T \cdot x_{(t)}^{(i)} + b_g), \tag{5}$$

$$c_{(t)} = f_{(t)} \otimes c_{(t-1)} + i_{(t)} \otimes g_{(t)}, \tag{6}$$

$$y_{(t)} = h_{(t)} = o_{(t)} \otimes \tanh(c_{(t)}). \tag{7}$$

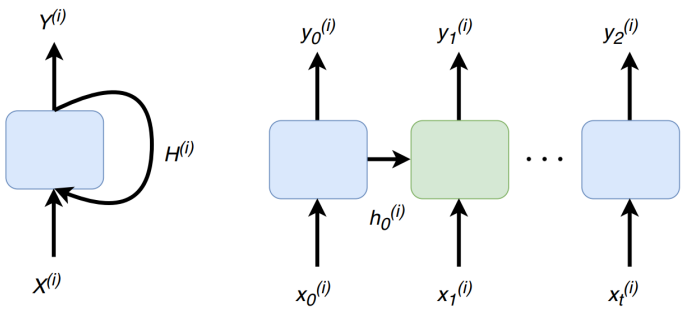


**Figure 4.** A recurrent layer and its unrolled through time representation. A multi-temporal, multi-spectral pixel $X^{(i)}$ is made by a sequence of time steps, $x_t^{(i)}$, that along the previous output $h^{(i)}$ feed the next iteration of the network.

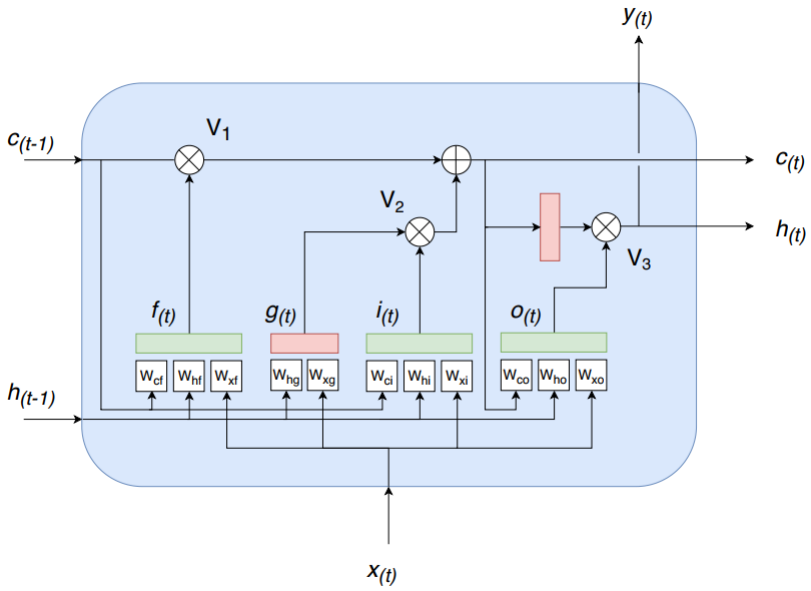


**Figure 5.** LSTM with peephole connections. A time step $t$ of a multi-spectral pixel $x_t^{(i)}$ is processed by the memory cell which decides what to add and forget in the long-term state $c_{(t)}$ and what discard for the present state $y_t^{(i)}$.

In conclusion, multi-temporal, multi-spectral pixel $X^{(i)}$ is processed by the first layer of LSTM peephole cells obtaining a cumulative output $Y^{(i)}_{(lstm)}$. Finally, a TimeDistributedDense layer is applied which executes simply a Dense function across every output over time, using the same set of weights, preserving the multidimensional nature of the processed data Equation (8). In Figure 6 is presented a graphical representation of the first layer of the network. LSTM cells extract temporal representations from input samples $X^{(i)}$ with $x_t^{(i)} \in \mathbb{R}^{(1*b)}$ as columns. The output matrix $Y^{(i)}_{(lstm)}$ feeds the subsequent TimeDistributedDense layer:

$$F_{timeD}(F_{lstm}(X^{(i)})) = (W \cdot Y^{(i)}_{(lstm)} + B). \tag{8}$$

### 4.1.2. Temporal Patterns Extraction

The first set of layers extract a 2-dimensional tensor $Y^{(i)}_{(timeD)}$ for each instance. In the second operation, after a simple reshaping operation in order to increase the dimensionality of the input tensor from two to three and being able to apply the following operations, we map each of these 3-dimensional array $\mathbf{Y}^{(i)}_{(reshape)}$ into a higher-dimensional space. That is accomplished by two convolutional operations, built on top of each other, that hierarchically apply learned filters, extracting gradually more abstract representations. More formally, the temporal patterns extraction is expressed, for example, for the first convolutional layer, as an operation $F_{conv1}$

$$F_{conv1}(F_{timeD}(F_{lstm}(X^{(i)}))) = \max(0, W_1 * \mathbf{Y}^{(i)}_{(reshape)} + B_1), \tag{9}$$

where $W_1$ and $B_1$ represent filters and biases, respectively, and '$*$' is the convolutional operation. $W_1$, contains $n_1$ filters with kernel dimension $f_1 \times f_1 \times c$, where $f_1$ is the spatial size of a filter and $c$ is the number of input channels. As common for CNN, the rectified linear unit (ReLU), max(0,x), has been chosen as the activation function for both layers units. In Figure 7 is depicted a graphical scheme of this section of the model. So, summarizing, output matrix $Y^{(i)}_{(timeD)}$ of the TimeDistributedDense layer, after adding an extra dimension, feeds a stack of two convolutional networks that progressively reduce the first two dimensions, gradually extracting higher-level representations and generating high

dimensional arrays. Moreover, being the $n_1$ and $n_2$ filters shared across all units, the same operation carried out with a similarly-sized dense fully connected layers would require a much greater number of parameters and computational power. Instead, the synergy of RNN and CNN opens the possibility to elaborate the overall temporal canvas in an optimal and efficient way.

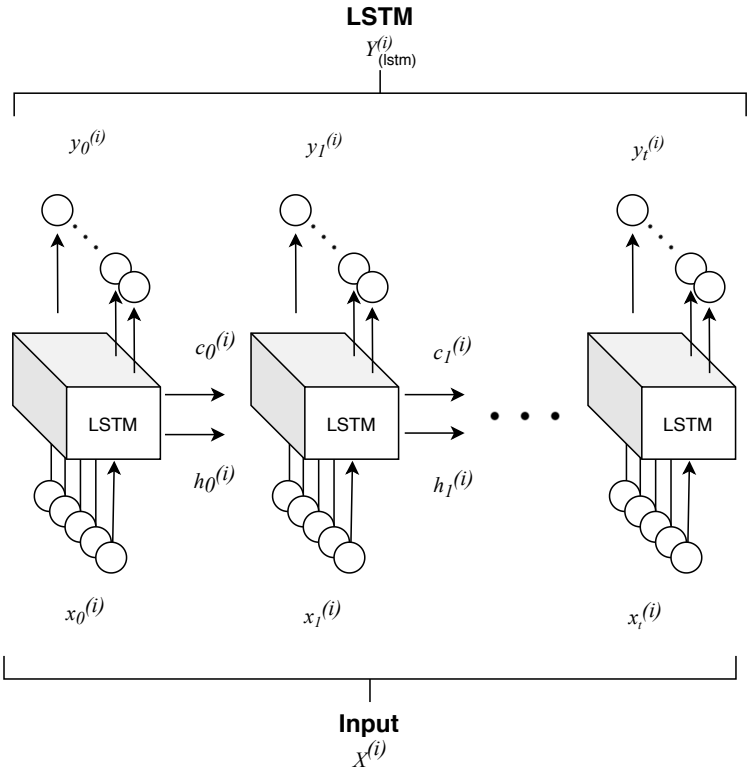


**Figure 6.** Pixel R-CNN first layer for time correlations extraction. Peephole LSTM cells extract temporal representations from input instances $X^{(i)} \in \mathbb{R}^{t*b}$. The output matrix $Y^{(i)}_{(lstm)}$ feeds a TimeDistributedDense layer, that preserves the multidimensional nature of the processed data extracting multi-spectral patterns.

### 4.1.3. Multiclass Classification

In the last stage of the network, the extracted feature tensor $\mathbf{Y}^{(i)}_{(conv2)}$, after removing the extra dimensions with a simple flatten operation, is mapped to a probability distribution consisting of $K$ probabilities, where $K$ is equal to the number of classes. This is achieved by a weighted sum followed by a softmax activation function:

$$\hat{p}_k = \sigma(s(x))_k = \frac{\exp s_k(x)}{\sum_{j=1}^{K} \exp s_j(x)} for\ j = 1, \ldots, K, \tag{10}$$

where $s(x) = W^T.y^{(i)}_{(flatten-conv2)} + B$ is a vector containing the scores of each class for the input vector $y^{(i)}_{(flatten-conv2)}$, that after the flatten operation is a 1-dimensional array. Weights W and bias B are learned, during the training process, in such a way to classify arrays of the high dimensional space into the K different classes. So, $\hat{p}_k$ is the estimated probability that the extracted feature vector $y^{(i)}_{(flatten-conv2)}$ belongs to class k given the scores of each class for that instance.

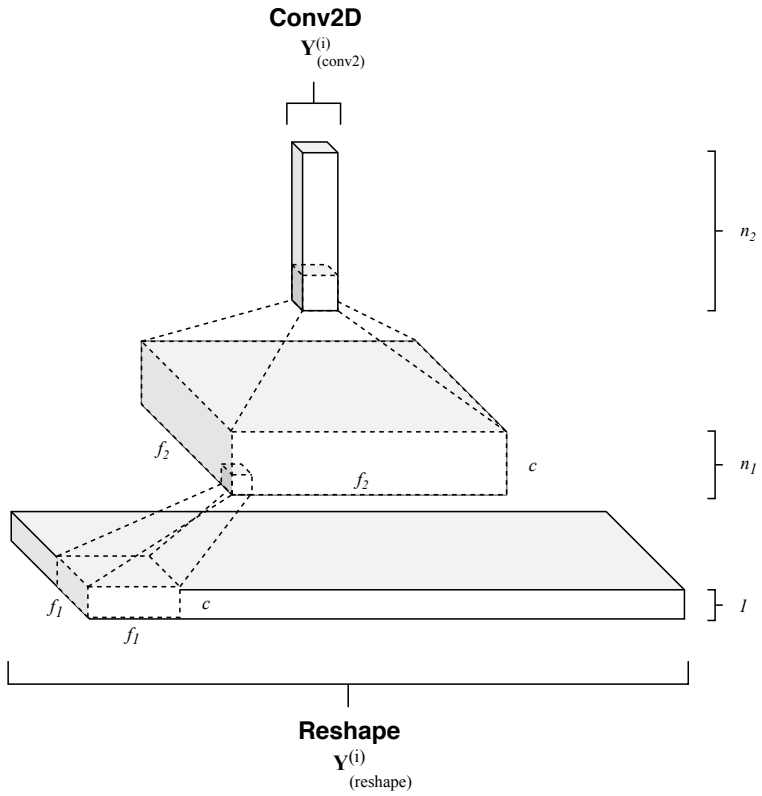


**Figure 7.** Pixel R-CNN convolutional layers. Firstly, output from TimeDistributedDense layer $Y^{(i)}_{(timeD)}$ is reshaped in a 3-dimensional tensor $\mathbf{Y}^{(i)}_{(reshape)}$ and then it feeds a stack of two convolutional layers that progressively reduce the first two dimensions, gradually extracting higher-level representations.

### 4.2. Training

Learning the overall mapping function F requires the estimation of all network parameters $\Theta$ of the three different model parts. This is simply achieved through minimizing the loss between each pixel class prediction $F(X^{(i)})$ and the corresponding ground truth $y^{(i)}$ with a supervised learning strategy. So, given a data set with $n$ pixel samples $\{X_i\}$ and the respective true classes set $\{y_i\}$, we use categorical cross-entropy as the loss function:

$$J(\Theta) = -1/n \sum_{i=1}^{n} \sum_{k=1}^{K} y_k^{(i)} \log(\hat{p}_k^{(i)}), \tag{11}$$

where $y_k^{(i)}$ cancels all classes loss except for the true one. Equation (11) is minimized using AMSGrad optimizer [71], an adaptive learning rate method that modifies the basic ADAM optimizer [72] algorithm. The overall algorithm update rule without the debiasing step is:

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t, \tag{12}$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2, \tag{13}$$

$$\hat{v}_t = \max(\hat{v}_{t-1}, v_t), \tag{14}$$

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t + \epsilon}} m_t. \tag{15}$$

Equations (12) and (13) are the exponential decay of the gradient and gradient squared, respectively. Instead, with the Equation (14), keeping a higher $v_t$ term results in a much lower learning rate, $\eta$, fixing the exponential moving average and preventing to converge to a sub-optimal

point of the cost function. Moreover, we use a technique known as cosine aneling in order to cyclically vary the learning rate value between certain boundary values [73]. This value can be obtained with a preliminary training procedure, linearly increasing the learning rate while observing the value of the loss function. Finally, we employ, as only regularization methodology, "Dropout" [74] in the time representation stage, inserted between the LSTM and Time-Distributed layer. This simple tweak allows training a more robust and resilient to noise temporal patterns extraction stage. Indeed, forcing CNN to work without relying on certain temporal activations can greatly improve the abstraction of the generated representations distributing the knowledge across all available units.

## 5. Experimental Results and Discussion

We first processed raw data in order to create a set of **n** pixel samples $\mathbb{X} = \{\mathbf{X}_i\}$ with the related ground truth labels $\mathbb{Y} = \{y_i\}$. Then, in order to have a visual inspection of the data set, principal component analysis (PCA), one of the most popular dimensionality reduction algorithms, have been applied to project the training set onto a lower tri-dimensional hyperplane. Finally, quantitative and qualitative results are discussed with a detail description of the architecture settings.

### *5.1. Training Data*

Sample pixels require to be extracted from the raw data and then reordered to feed the devised architecture. Indeed, the first RNN stage requires data points to be collected in slices of time series.

So, we separated labeled pixels from raw data, and we divided them in chunks of data, forming a tri-dimensional tensor $\mathbf{X} \in \mathbb{R}^{i \times t \times b}$ for the successive pre-processing pipeline. In Figure 8, a visual representation of the data set tensor $\mathbf{X}$ generation, where fixing the first dimension $\mathbf{X}_{i,:,:}$ there are the individual pixel samples $X^{(i)}$ with $t = 9$ time steps and $b = 5$ spectral bands. It is worth to notice that the number of time steps and bands are completely an arbitrary choice dictated by the raw data availability.

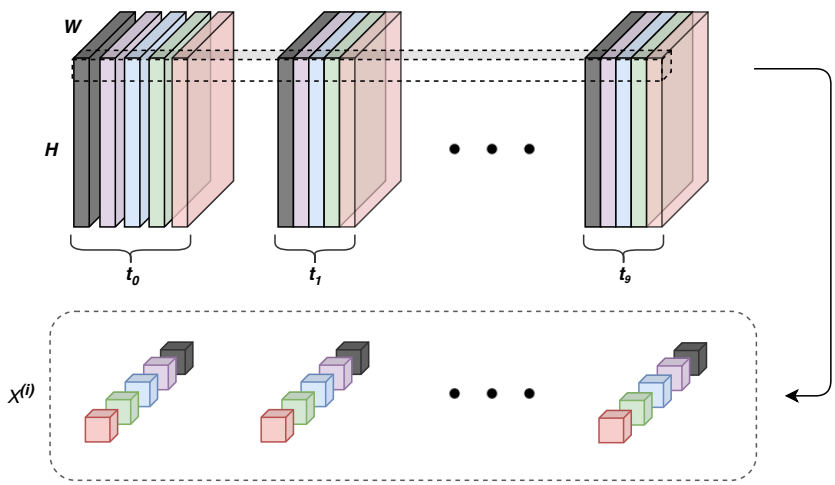


**Figure 8.** Overview of the tensor $\mathbf{X} \in \mathbb{R}^{i \times t \times b}$ generation. The first dimension $i$ represents the collected instances $X^{(i)}$, the second $t$ the different time steps, and finally the last one $b$ the five spectral bands, red, green, blue, near-infrared and NDVI. On the top, labeled pixels are extracted simultaneously, from all-time steps and bands starting from the raw satellite images. Then, $X_{i,j,k}$ are reshaped in order to set up the $X^{(i)} = \mathbf{X}_{i,:,:}$ of the data set tensor $\mathbf{X}$.

Subsequently, we adopted a simple pipeline of two steps to pre-process the data. Stratified sampling has been applied in order to divide the data set tensor $\mathbf{X}$, with shape (92116, 9, 5), in a training and test set. Due to the natural unbalanced number of instances per class present in the data set Table 3, this is an important step to preserve the same percentage in the two sets. After selecting a split percentage for the training of 60%, we obtained two tensors $\mathbf{X}_{train}$ and $\mathbf{X}_{test}$ with shape (55270, 9,

5) and (36846, 9, 5), respectively. Secondly, as common practice, to facilitate the training, we adopted standard scaling, $(x - \mu)/\sigma$, to normalize the two sets of data points.

**Table 3.** Land cover types contribution in the reference data.

| Class | Pixels | Percentage |
|---|---|---|
| Tomatoes | 3020 | 3.20% |
| Artificials | 9343 | 10.14% |
| Trees | 7384 | 8.01% |
| Rye | 4382 | 4.75% |
| Wheat | 12,826 | 13.92% |
| Soya | 5836 | 6.33% |
| Apple | 849 | 0.92% |
| Peer | 495 | 0.53% |
| Temp Grass | 1744 | 1.89% |
| Water | 2451 | 2.66% |
| Lucerne | 17,942 | 19.47% |
| Durum Wheat | 1188 | 1.28% |
| Vineyard | 6110 | 6.63% |
| Barley | 2549 | 2.76% |
| Maize | 15,997 | 17.37% |
| **Total** | 92,116 | 100% |

### 5.2. Dataset Visualization

To explore and visualize the generated set of points, we exploit Principle Component Analysis (PCA), reducing the high dimensionality of the data set. For this operation, we considered the components t and b as features of our data points. So, applying Singular Value Decomposition (SVD) and then selecting the first three principal components, $W_d = (c_1, c_2, c_3)$, it was possible to plot the different classes in a tri-dimensional space, having a visual representation of the projected data points. In Figure 9 the projected data points are plotted in tri-dimensional space. Except for water bodies, it is worth to point out how much intra-class variance is present. Indeed, most of the classes lay on more than one hyperplane, demonstrating the difficulty of the task and the studied data set. Finally, it was possible to analyze the explained variance ratio varying the number of dimensions. From Figure 10, it is worth to notice that approaching higher components, the explained variance trend stops growing fast. So, that can be considered as the intrinsic dimensionality of the data set. Due to this fact, it is reasonable to assume that reducing the number of time steps would not significantly affect the overall results.

### 5.3. Experimental Settings

In this section, we examine the settings of the final network architecture. The basic Pixel R-CNN model, shown in Figure 3, is the result of a careful design aimed at obtaining the best performance in terms of accuracy and computational cost. Indeed, the final model is a lightweight model with 30,936 trainable parameters (less than 1 MB), fast and more accurate than the existing state-of-the-art solutions. With the suggested approach, we employed only an RNN layer with 32 output units for each peephole LSTM cell randomly turned off, with a probability $p$ = 0.2, by a Dropout regularization operation. For all experiments, peephole LSTM has shown an improvement in overall accuracy around 0.8% over standard LSTM cells. Then, Time Distributed Dense transforms $Y^{(i)}_{(lstm)}$ in a $9 \times 9$ square matrix that feed a stack of two CNN layers with a number of features $n_1$ = 16 and $n_2$ = 32, respectively. The first layer as a filter size of $f_1$ = 3 and the second one $f_2$ = 7 producing a one-dimensional output array. Finally, a fully connected layer with SoftMax activation function maps $Y^{(i)}_{(conv2)}$ to the probability of the $K = 15$ different classes. Except for the final layer, we adopted ReLU as activation functions. To find the best training hyperparameters for the optimizer, we used 10% of the training set to perform a random search evaluation, with few epochs, in order to select the most promising parameters. Then,

after this first preliminary phase, the analysis has been focused only on the value of the most promising hyperparameter, fine-tuning them with a grid search strategy.

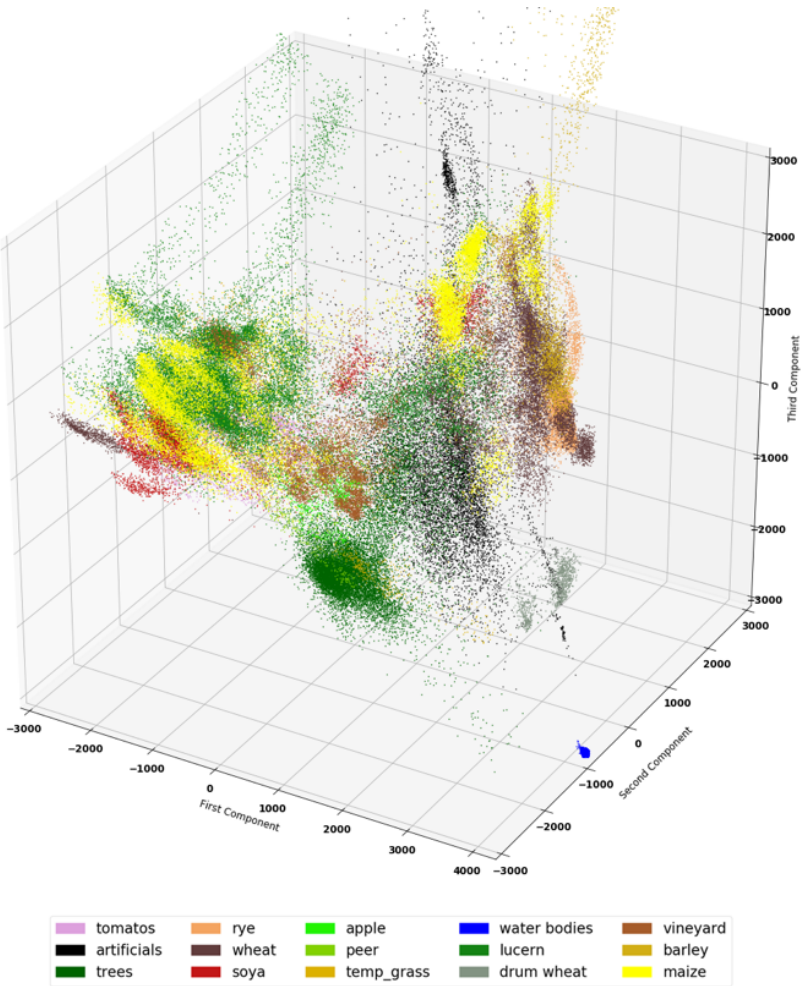


**Figure 9.** Visual representation of the data points projected in the tri-dimensional space using PCA. The three principal components took into account preserve 64.5% of the original data set variance.

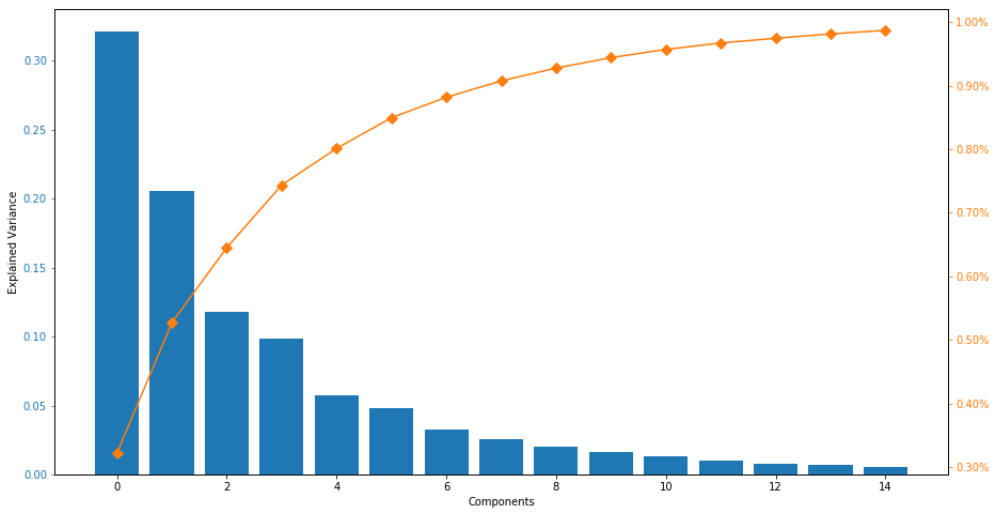


**Figure 10.** Pareto chart of the explained variance as a function of the number of components.

So, for the AMSGrad optimizer, we set $\beta_1 = 0.86$, $\beta_2 = 0.98$, and $\epsilon = 10^{-9}$. Finally, as previously introduced, with a preliminary procedure, we linearly increased the learning rate of $\eta$ while observing the value of the loss function to estimate the initial value of this important hyperparameter. In conclusion, we fed our model with more than 62,000 samples for 150 epochs with a batch size of 128 while cyclically varying the learning rate value with a cosine aneling strategy. All tests have been carried out with the TensorFlow framework on a workstation with 64 GB RAM, Intel Core i7-9700K CPU, and an Nvidia 2080 Ti GPU.

### 5.4. Classification

Performance of the classifier was evaluated by user's accuracy (UA), producer's accuracy (PA), overall accuracy (OA), and the kappa coefficient (K) shown in the confusion matrix see Table 4, which is the most common metric that has been used for classification tasks [19,75–77]. Overall accuracy indicates the overall performance of our proposed Pixel R-CNN architecture by calculating a ratio between the correctly classified total number of pixels and total ground truth pixels for all classes. The diagonal elements of the matrix represent the pixels that were classified correctly for each class. Individual class accuracy was calculated by dividing the number of correctly classified pixels in each category by the total number of pixels in the corresponding row called User's accuracy, and columns called Producer's accuracy. PA indicates the probability that a certain crop type on the ground is classified as such.

UA represents the probability that a pixel classified in a given class belongs to that class. Our proposed pixel-based Pixel R-CNN method achieved OA = 96.5% and Kappa = 0.914 with 15 number of classes for a diverse large scale area, which exhibits significant improvement as compared to other mainstream methods. Water bodies and trees stand highest in terms of UA with 99.1% and 99.3%, respectively. That is mainly due to intra-class variability and the minor change of NIR band reflectance over time, which was easily learned by our Pixel R-CNN. Most of the classes, including the major types of crops such as Maize, Wheat, Lucerne, Vineyard, Soya, Rye, and Barley, were classified with more than 95% UA. Grassland being the worst class, which was classified with the PA = 65% and UA = 63%. The major confusion of grassland class was with Lucerne and Vineyard. It is worth mentioning that the Artificial class, which belongs to roads, buildings, urban areas, represents the mixed nature of pixel reflectances and was accurately detected with UA = 97% and PA = 99%.

For class Apple, obtained PA was 86% while UA = 68%, which shows that 86% of the ground truth pixels were identified as Apple, but only 68% of the pixels classified as Apple in the classification actually belonged to class Apple. Some Pixels (see Table 4) belongs to Peer and Vineyard were mistakenly classified as Apple.

The final classified map is shown in Figure 11 with the example of the zoomed part and the actual RGB image. To the best of our knowledge, a multi-temporal benchmark dataset is not available to compare classification approaches on equal footings. There are some data sets available online for crop classification without having ground truth of other land cover types such as Trees, Artificial land (build ups), Water bodies, Grassland. Therefore it is difficult to compare classification approaches on equal footings. Indeed, a direct quantitative comparison of the classification performed in these studies is difficult due to various dependencies such as the number of evaluated ground truth samples, the extent of the considered study area, and the number of classes to be evaluated. Nonetheless, we provided an overview of recent studies and their performances of the study domain by their applied approaches, the number of considered classes, used sensors, and achieved overall accuracy in Table 5. Hao et al. [30], achieved 89% OAA by using RF classifier on the extracted phenological features from MODIS time-series data. They determined that good classification accuracies can be achieved with handcrafted features and classification algorithms if the temporal resolution of the data is sufficient. Though, the MODIS sensor data is not suitable for classification of the areas of large homogeneous regions due to its low spatial resolution (500 m). Conrad et al. [78] used high spatial resolution data from the RapidEye sensor and achieved 90% OAA for nine considered classes.

In [76], features from optical and SAR were extracted and used by the committee of neural networks of multilayer perceptrons to classify a diverse agriculture region considerably. Recurrent encoders have been employed in [79] to classify a large area for 17 considered classes using high spatial resolution (10 m) sentinel-2 data and achieved 90% OAA, which proved that recurrent encoders are useful to capture the temporal information of spectral features that leads to higher accuracy. Voulo et al. [77] also used sentinel-2 data and achieved a maximum 95% classification accuracy using RF classifier but nine classes were considered.

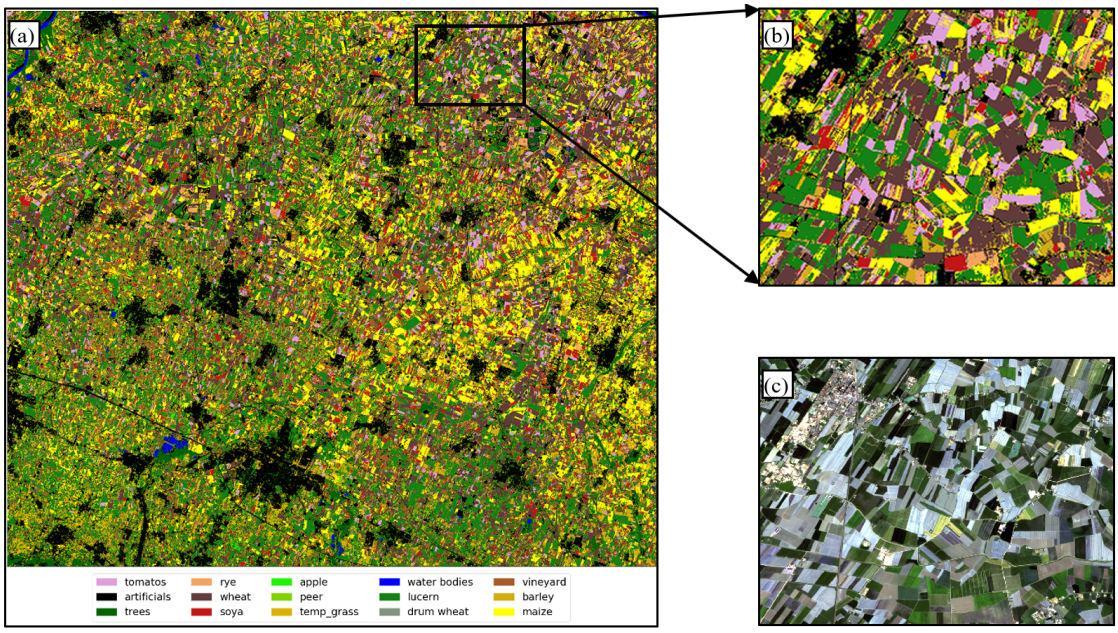


**Figure 11.** (**a**) Final classified map using Pixel R-CNN, (**b**) zoomed in region of the classified map, and (**c**) Raw Sentinel-2 RGB composite of the zoomed region.

In conclusion, it is interesting to notice neuron activation inside the network during the classification process. Indeed, it is possible to plot unit values when the network receives specific inputs and compare how model behaviors change. In Figure 12 four samples, belonging to the same class "artificials", feed the model creating individual activations in the input layer. Even if they all belong to the same class, the four instances took into account present a noticeable variance. Either the spectral features in a specific time instance (rows) or their temporal variation (columns) present different patterns that make them difficult to classify. However, already after the first LSTM layer with the TimeDistributedDense block, the resulting $9 \times 9$ matrices $Y_{(timeD)}$ have a clear pattern that can be used by the following layers to classify the different instances in their respective classes. So, the network during the training process learns to identify specific temporal schemes, that allows making strong distributed and disentangle representations.

**Table 4.** Obtained confusion matrix.

| Ground Truth | Classified Classes | | | | | | | | | | | | | | | Total | PA |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | **TM** | **AR** | **TR** | **RY** | **WH** | **SY** | **AP** | **PR** | **GL** | **WT** | **LN** | **DW** | **VY** | **BL** | **MZ** | | |
| Tomatoes (TM) | **1096** | 0 | 0 | 0 | 4 | 11 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1111 | 98% |
| Artificial (AR) | 0 | **3752** | 8 | 1 | 2 | 0 | 2 | 1 | 9 | 9 | 12 | 2 | 6 | 0 | 4 | 3808 | 99% |
| Trees (TR) | 0 | 31 | **2967** | 1 | 0 | 0 | 0 | 3 | 10 | 0 | 17 | 0 | 2 | 0 | 0 | 3031 | 98% |
| Rye (RY) | 0 | 1 | 0 | **1960** | 25 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 5 | 0 | 1991 | 98% |
| Wheat (WH) | 38 | 7 | 0 | 221 | **4981** | 6 | 0 | 0 | 10 | 0 | 14 | 1 | 2 | 38 | 42 | 5360 | 93% |
| Soya (SY) | 3 | 0 | 0 | 0 | 3 | **1226** | 0 | 0 | 0 | 0 | 11 | 0 | 3 | 0 | 41 | 1287 | 95% |
| Apple (AP) | 0 | 0 | 0 | 0 | 0 | 0 | **142** | 0 | 0 | 0 | 2 | 0 | 21 | 0 | 0 | 165 | 86% |
| Peer (PR) | 0 | 0 | 11 | 0 | 0 | 0 | 27 | **124** | 0 | 0 | 0 | 0 | 6 | 0 | 0 | 168 | 73% |
| Grassland (GL) | 0 | 39 | 3 | 7 | 0 | 1 | 0 | 0 | **239** | 0 | 72 | 0 | 3 | 0 | 4 | 368 | 65% |
| Water (WT) | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | **906** | 0 | 0 | 0 | 0 | 0 | 906 | 100% |
| Lucerne (LN) | 0 | 0 | 0 | 2 | 0 | 2 | 0 | 0 | 48 | 0 | **7250** | 0 | 26 | 0 | 10 | 7338 | 98% |
| Durum.Wheat (W) | 0 | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 0 | 0 | **322** | 0 | 0 | 0 | 328 | 98% |
| Vineyard (VY) | 11 | 7 | 4 | 4 | 11 | 1 | 50 | 1 | 21 | 0 | 93 | 0 | **2139** | 0 | 7 | 2349 | 91% |
| Barley (BL) | 0 | 1 | 0 | 2 | 24 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | **817** | 0 | 846 | 96% |
| Maize (MZ) | 17 | 14 | 0 | 0 | 10 | 24 | 0 | 3 | 10 | 0 | 16 | 1 | 6 | 0 | **7689** | 7790 | 99% |
| Total | 1165 | 3856 | 2993 | 2198 | 5060 | 1271 | 221 | 132 | 350 | 915 | 7488 | 326 | 2214 | 860 | 7797 | | |
| UA | 94% | 97% | 99% | 89% | 98% | 96% | 64% | 93% | 68% | 99% | 96% | 99% | 96% | 95% | 98% | | |

**Table 5.** An overview and performance of recent studies.

| Study | Details | | | | |
|---|---|---|---|---|---|
| | **Sensor** | **Features** | **Classifier** | **Accuracy** | **Classes** |
| Our | Sentinel-2 | BOA Reflectances | Pixel R-CNN | 96.50% | 15 |
| Rußwurm and Körner [78], 2018 | Sentinel-2 | TOA Reflectances | Recurrent Encoders | 90% | 17 |
| Skakun et al. [77], 2016 | Radarsat-2 + Landsat-8 | Optical+SAR | NN and MLPs | 90% | 11 |
| Conrad et al. [76], 2014 | RapidEye | Vegetation Indices | RF and OBIA | 86% | 9 |
| Vuolo et al. [80], 2018 | Sentinel-2 | Optical | RF | 91–95% | 9 |
| Hao et al. [30], 2015 | MODIS | Stat + phenological | RF | 89% | 6 |
| J.M. Peña-Barragán [81], 2011 | ASTER | Vegetation Indices | OBIA+DT | 79% | 13 |

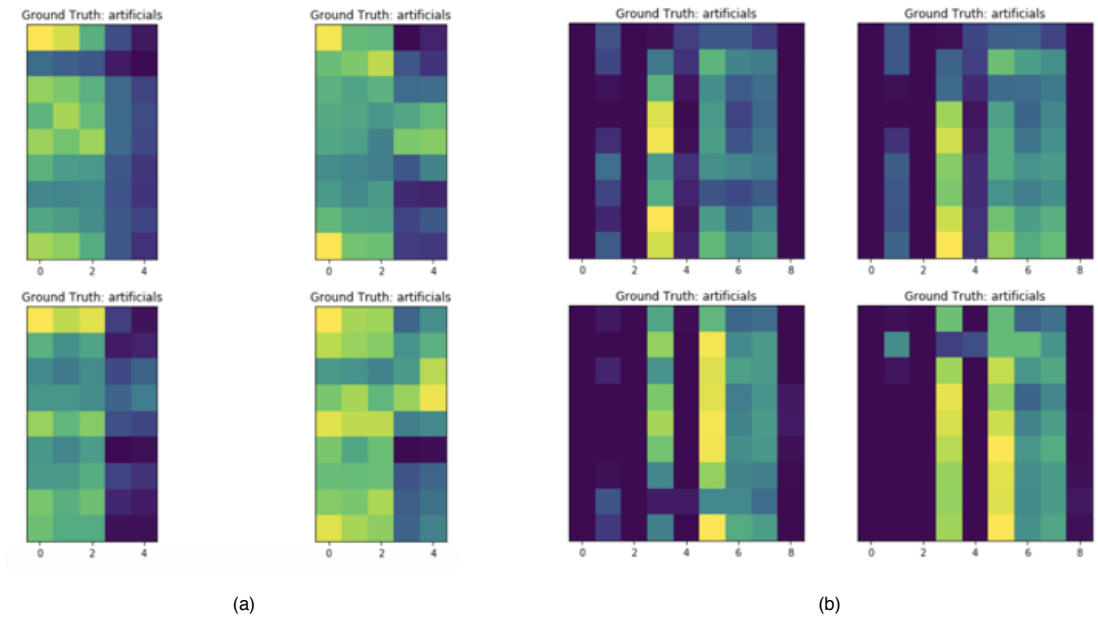


**Figure 12.** Visual representation of the activation of the internal neurons of Pixel R-CNN, where darker color are values close to zero and vice versa. (**a**). four samples of the same class "artificials", (**b**). related activations inside the network at the output of the TimeDistributedDense layer $Y_{(timeD)}$. It is interesting to notice how the four inputs are pretty different from each other, but the network representations already at this level are similar.

### *5.5. Non Deep Learning Classifiers*

We tried four other traditional classifiers on the same dataset for comparison, which are Support Vector Machine (SVM), Kernal SVM, Random Forest (RF), and XGBoost. These are well-known classifiers for their high performances and also considered as baseline models in classification tasks [74]. SVM can perform nonlinear classification using kernel functions by separating hyperplanes. A widely used RF classifier is an ensemble of decision trees based on the bagging approach [82,83]. XGBoost is state of the art classifier based on gradient boosting model of decision trees, which attracted much attention in the machine learning community. RF and SVM have been widely used in remote sensing applications [29,30,46]. Each classifier involves hyperparameters that need to be tuned at the time of classification model development.

We followed the "random search" approach to optimize major hyperparameters [84]. Best values of hyperparameters were selected based on classification accuracy achieved for the validation set, and are highlighted with bold letters in Table 6. Further details about hyper parameters and achieved overall accuracy (OA) for SVM, Kernal SVM, RF, and XGBoost are reported in Table 6 From these non-deep learning classifiers, SVM stands highest with OA = 79.6% while RF, Kernel SVM, and XGBoost achieved 77.5%, 76.8%, and 77.2% respectively. From the results presented in Table 6, our proposed Pixel R-CNN based classifier achieved OA = 96.5%, which is far better results than the non deep learning classifiers. Learning temporal and spectral correlations from multi-temporal images considering large data set is challenging for traditional non-deep learning techniques. The introduction of deep learning models in the remote sensing domain brought more flexibility to exploit temporal features in such a way that it can increase the amount of information to gain much better and reliable results for classification tasks.

**Table 6.** Comparison of Pixel R-CNN with non-deep learning classifiers.

| Model | Parameters | OA |
|---|---|---|
| SVM | C: 0.01, 0.1, **1**, 10, 100, 1000<br>Kernel: linear | 79.50% |
| Kernel SVM | C: 0.01, 0.1, **1**, 10, 100, 1000<br>Kernel: **rbf**<br>Gamma: **0.1**, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8 | 76.20% |
| Random Forest | n_estimators: 10, 20, 100, 200, **500** max_depth: **5**, 10, 15, 30 min_samples_split: 3, **5**, 10, 15, 30 min_samples_leaf: 1, 3, **5**, 10 | 77.90% |
| XGBoost | learning_rate: **0.01**, 0.02, 0.05, 0.1<br>gamma: 0.05, **0.1**, 0.5, 1<br>max_depth: 3, **7**, 9, 20, 25<br>min_child_weight: 1, 5, 7, **9**<br>subsamples: 0.5, **0.7**, 1<br>colsample_bytree: **0.5**, 0.7, 1<br>reg_labda: 0.01, 0.1, **1**<br>reg_alpha: 0, 0.1, 0.5, **1** | 77.60% |
| **Pixel R-CNN** | Mentioned in experimental settings | **96.50%** |

## 6. Conclusions

In this study, we developed a novel deep learning model with Recurrent and Convolutional Neural Network called Pixel R-CNN to perform Land Cover and Crop Classification by using multitemporal decametric sentinel-2 imagery of central north part of Italy. Our proposed Pixel R-CNN based architecture exhibits significant improvement as compared to other mainstream methods by achieving 96.5% overall accuracy with kappa = 0.914 for 15 number of classes. We also tested widely used non-deep learning classifiers such as SVM, RF, SVM kernel, and XGBoost to compare with our proposed classifier and revealed that these methods are less effective, especially when the temporal featureq extraction is the key to increase classification accuracy. The main advantage of our architecture is the capability of automated feature extraction by learning time correlation of multiple images, which reduces manual feature engineering and modeling crops phenological stages.

**Author Contributions:** Conceptualization, A.K.; Formal analysis, A.K.; Funding acquisition, M.C.; Investigation, A.K. and M.C.; Methodology, V.M.; Project administration, M.C.; Resources, M.C.; Software, V.M.; Supervision, M.V.; Visualization, A.K.; Writing—original draft, V.M. and A.K.; Writing—review & editing, A.K. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Acknowledgments:** This work has been developed with the contribution of the Politecnico di Torino Interdepartmental Centre for Service Robotics PIC4SeR (https://pic4ser.polito.it) and SmartData@Polito (https://smartdata.polito.it).

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Gomez, C.; White, J.C.; Wulder, M.A. Optical remotely sensed time series data for land cover classification: A review. *ISPRS J. Photogramm. Remote Sens.* **2016**, *116*, 55–72. [CrossRef]
2. Wu, W.-B.; Yu, Q.-Y.; Peter, V.H.; You, L.-Z.; Yang, P.; Tang, H.-J. How Could Agricultural Land Systems Contribute to Raise Food Production Under Global Change? *J. Integr. Agric.* **2014**, *13*, 1432–1442. [CrossRef]
3. Jin, Z.; Azzari, G.; You, C.; Di Tommaso, S.; Aston, S.; Burke, M.; Lobell, D.B. Smallholder maize area and yield mapping at national scales with Google Earth Engine. *Remote Sens. Environ.* **2019**, *228*, 115–128. [CrossRef]

4. Yang, P.; Wu, W.-B.; Tang, H.-J.; Zhou, Q.-B.; Zou, J.-Q.; Zhang, L. Mapping Spatial and Temporal Variations of Leaf Area Index for Winter Wheat in North China. *Agric. Sci. China* **2007**, *6*, 1437–1443. [CrossRef]
5. Wang, L.A.; Zhou, X.; Zhu, X.; Dong, Z.; Guo, W. Estimation of biomass in wheat using random forest regression algorithm and remote sensing data. *Crop. J.* **2016**, *4*, 212–219. [CrossRef]
6. Matton, N.; Canto, G.; Waldner, F.; Valero, S.; Morin, D.; Inglada, J.; Defourny, P. An automated method for annual cropland mapping along the season for various globally-distributed agrosystems using high spatial and temporal resolution time series. *Remote Sens.* **2015**, *7*, 13208–13232. [CrossRef]
7. Guan, K.; Berry, J.A.; Zhang, Y.; Joiner, J.; Guanter, L.; Badgley, G.; Lobell, D.B. Improving the monitoring of crop productivity using spaceborne solar-induced fluorescence. *Glob. Chang. Biol.* **2016**, *22*, 716–726. [CrossRef]
8. Battude, M.; Al Bitar, A.; Morin, D.; Cros, J.; Huc, M.; Marais Sicre, C.; Le Dantec, V.; Demarez, V. Estimating maize biomass and yield over large areas using high spatial and temporal resolution Sentinel-2 like remote sensing data. *Remote Sens. Environ.* **2016**, *184*, 668–681. [CrossRef]
9. Huang, J.; Tian, L.; Liang, S.; Ma, H.; Becker-Reshef, I.; Huang, Y.; Su, W.; Zhang, X.; Zhu, D.; Wu, W. Improving winter wheat yield estimation by assimilation of the leaf area index from Landsat TM and MODIS data into the WOFOST model. *Agric. For. Meteorol.* **2015**, *204*, 106–121. [CrossRef]
10. Toureiro, C.; Serralheiro, R.; Shahidian, S.; Sousa, A. Irrigation management with remote sensing: Evaluating irrigation requirement for maize under Mediterranean climate condition. *Agric. Water Manag.* **2017**, *184*, 211–220. [CrossRef]
11. Yu, Q.; Shi, Y.; Tang, H.; Yang, P.; Xie, A.; Liu, B.; Wu, W. eFarm: A Tool for Better Observing Agricultural Land Systems. *Sensors* **2017**, *17*, 453. [CrossRef] [PubMed]
12. Boryan, C.; Yang, Z.; Mueller, R.; Craig, M. Monitoring US agriculture: the US department of agriculture, national agricultural statistics service, cropland data layer program. *Geocarto Int.* **2011**, *26*, 341–358. [CrossRef]
13. Xie, Y.; Sha, Z.; Yu, M. Remote sensing imagery in vegetation mapping: A review. *Plant Ecol.* **2008**, *1*, 9–23. [CrossRef]
14. Johnson, D.M. A comprehensive assessment of the correlations between field crop yields and commonly used MODIS products. *Int. J. Appl. Earth Obs. Geoinf.* **2016**, *52*, 65–81. [CrossRef]
15. Senf, C.; Leitão, P.J.; Pflugmacher, D.; van der Linden, S.; Hostert, P. Mapping land cover in complex Mediterranean landscapes using Landsat: Improved classification accuracies from integrating multi-seasonal and synthetic imagery. *Remote Sens. Environ.* **2015**, *156*, 527–536. [CrossRef]
16. Jin, H.; Li, A.; Wang, J.; Bo, Y. Improvement of spatially and temporally continuous crop leaf area index by integration of CERES-Maize model and MODIS data. *Eur. J. Agron.* **2016**, *78*, 1–12. [CrossRef]
17. Liaqat, M.U.; Cheema, M.J.M.; Huang, W.; Mahmood, T.; Zaman, M.; Khan, M.M. Evaluation of MODIS and Landsat multiband vegetation indices used for wheat yield estimation in irrigated Indus Basin. *Comput. Electron. Agric.* **2017**, *138*, 39–47. [CrossRef]
18. Senf, C.; Pflugmacher, D.; Heurich, M.; Krueger, T. A Bayesian hierarchical model for estimating spatial and temporal variation in vegetation phenology from Landsat time series. *Remote Sens. Environ.* **2017**, *194*, 155–160. [CrossRef]
19. Kussul, N.; Lemoine, G.; Gallego, F.J.; Skakun, S.V.; Lavreniuk, M.; Shelestov, A.Y. Parcel-based crop classification in ukraine using landsat-8 data and sentinel-1A data. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2016**, *9*, 2500–2508. [CrossRef]
20. Xiong, J.; Thenkabail, P.S.; Gumma, M.K.; Teluguntla, P.; Poehnelt, J.; Congalton, R.G.; Thau, D. Automated cropland mapping of continental Africa using Google Earth Engine cloud computing. *ISPRS J. Photogramm. Remote Sens.* **2017**, *126*, 225–244. [CrossRef]
21. Yan, L.; Roy, D.P. Improved time series land cover classification by missing-observation-adaptive nonlinear dimensionality reduction. *Remote Sens. Environ.* **2015**, *158*, 478–491. [CrossRef]
22. Xiao, J.; Wu, H.; Wang, C.; Xia, H. Land Cover Classification Using Features Generated From Annual Time-Series Landsat Data. *IEEE Geosci. Remote Sens. Lett.* **2018**, *15*, 739–743. [CrossRef]
23. Khaliq, A.; Peroni, L.; Chiaberge, M. Land cover and crop classification using multitemporal Sentinel-2 images based on crops phenological cycle. In *IEEE Workshop on Environmental, Energy, and Structural Monitoring Systems (EESMS)*; IEEE: Piscataway, NJ, USA, 2018; pp. 1–5.

24. Gallego, J.; Craig, M.; Michaelsen, J.; Bossyns, B.; Fritz, S. *Best Practices for Crop Area Estimation with Remote Sensing*; Joint Research Center: Ispra, Italy, 2008.
25. Zhou, F.; Zhang, A.; Townley-Smith, L. A data mining approach for evaluation of optimal time-series of MODIS data for land cover mapping at a regional level. *ISPRS J. Photogramm. Remote Sens.* **2013**, *84*, 114–129. [CrossRef]
26. Arvor, D.; Jonathan, M.; Meirelles, M.S.P.; Dubreuil, V.; Durieux, L. Classification of MODIS EVI time series for crop mapping in the state of Mato Grosso, Brazil. *Remote Sens.* **2011**, *32*, 7847–7871. [CrossRef]
27. Zhong, L.; Gong, P.; Biging, G.S. Phenology-based crop classification algorithm and its implications on agricultural water use assessments in California's Central Valley. *Photogramm. Eng. Remote Sens.* **2012**, *78*, 799–813. [CrossRef]
28. Wardlow, B.D.; Egbert, S.L. A comparison of MODIS 250-m EVI and NDVI data for crop mapping: A case study for southwest Kansas. *Int. J. Remote Sens.* **2012**,*31*, 805–830. [CrossRef]
29. Löw, F.; Michel, U.; Dech, S.; Conrad, C. Impact of feature selection on the accuracy and spatial uncertainty of per-field crop classification using support vector machines. *ISPRS J. Photogramm. Remote Sens.* **2013**, *85*, 102–119. [CrossRef]
30. Hao, P.; Zhan, Y.; Wang, L.; Niu, Z.; Shakir, M. Feature selection of time series MODIS data for early crop classification using random forest: A case study in Kansas, USA. *Remote Sens.* **2015**, *7*, 5347–5369. [CrossRef]
31. Blaschke, T. Object based image analysis for remote sensing. *ISPRS J. Photogramm. Remote Sens.* **2010**, *65*, 2–16. [CrossRef]
32. Novelli, A.; Aguilar, M.A.; Nemmaoui, A.; Aguilar, F.J.; Tarantino, E. Performance evaluation of object based greenhouse detection from Sentinel-2 MSI and Landsat 8 OLI data: A case study from Almería (Spain). *Int. J. Appl. Earth Obs. Geoinf.* **2016,** *52*, 403–411. [CrossRef]
33. Long, J.A.; Lawrence, R.L.; Greenwood, M.C.; Marshall, L.; Miller, P.R. Object-oriented crop classification using multitemporal ETM+ SLC-off imagery and random forest. *GIsci. Remote Sens.* **2013**, *50*, 418–436. [CrossRef]
34. Li, Q.; Wang, C.; Zhang, B.; Lu, L. Object-based crop classification with Landsat-MODIS enhanced time-series data. *Remote Sens.* **2015**, *7*, 16091–16107. [CrossRef]
35. Walker, J.J.; De Beurs, K.M.; Wynne, R.H. Dryland vegetation phenology across an elevation gradient in Arizona, USA, investigated with fused MODIS and Landsat data. *Remote Sens. Environ.* **2014**, *144*, 85–97. [CrossRef]
36. Walker, J.J.; De Beurs, K.M.; Henebry, G.M. Land surface phenology along urban to rural gradients in the US Great Plains. *Remote Sens. Environ.* **2015**, *165*, 42–52. [CrossRef]
37. Simonneaux, V.; Duchemin, B.; Helson, D.; Er-Raki, S.; Olioso, A.; Chehbouni, A.G. The use of high-resolution image time series for crop classification and evapotranspiration estimate over an irrigated area in central Morocco. *Int. J. Remote Sens.* **2008**, *29*, 95–116. [CrossRef]
38. Shao, Y.; Lunetta, R.S.; Wheeler, B.; Iiames, J.S.; Campbell, J.B. An evaluation of time-series smoothing algorithms for land-cover classifications using MODIS-NDVI multi-temporal data. *Remote Sens. Environ.* **2016**, *174*, 258–265. [CrossRef]
39. Galford, G.L.; Mustard, J.F.; Melillo, J.; Gendrin, A.; Cerri, C.C.; Cerri, C.E. Wavelet analysis of MODIS time series to detect expansion and intensification of row-crop agriculture in Brazil. *Remote Sens. Environ.* **2008**, *112*, 576–587. [CrossRef]
40. Funk, C.; Budde, M.E. Phenologically-tuned MODIS NDVI-based production anomaly estimates for Zimbabwe. *Remote Sens. Environ.* **2009**, *113*, 115–125. [CrossRef]
41. Siachalou, S.; Mallinis, G.; Tsakiri-Strati, M. A hidden Markov models approach for crop classification: Linking crop phenology to time series of multi-sensor remote sensing data. *Remote Sens.* **2015**, *7*, 3633–3650. [CrossRef]
42. Xin, Q.; Broich, M.; Zhu, P.; Gong, P. Modeling grassland spring onset across the Western United States using climate variables and MODIS-derived phenology metrics. *Remote. Sens. Environ.* **2015**, *161*, 63–77. [CrossRef]
43. Gonsamo, A.; Chen, J.M. Circumpolar vegetation dynamics product for global change study. *Remote. Sens. Environ.* **2016**, *182*, 13–26. [CrossRef]
44. Dannenberg, M.P.; Song, C.; Hwang, T.; Wise, E.K. Empirical evidence of El Niño–Southern Oscillation influence on land surface phenology and productivity in the western United States. *Remote Sens. Environ.* **2015**, *159*, 167–180. [CrossRef]

45. Khatami, R.; Mountrakis, G.; Stehman, S.V. A meta-analysis of remote sensing research on supervised pixel-based land-cover image classification processes: General guidelines for practitioners and future research. *Remote Sens. Environ.* **2016**, *177*, 89–100. [CrossRef]
46. Gislason, P.O.; Benediktsson, J.A.; Sveinsson, J.R. Random forests for land cover classification. *Pattern Recognit. Lett.* **2006**, *27*, 294–300. [CrossRef]
47. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444. [CrossRef] [PubMed]
48. Sutskever, I.; Vinyals, O.; Le, Q.V. Sequence to sequence learning with neural networks. In Proceedings of the Twenty-Eighth Conference on Neural Information Processing Systems, Montreal, QC, Canada, 3–8 December 2014; pp. 3104–3112.
49. Gers, F.A.; Schmidhuber, J.; Cummins, F. Learning to forget: Continual prediction with LSTM. *Neural Comput.* **2000**, 850–855. [CrossRef]
50. Chung, J.; Gulcehre, C.; Cho, K.; Bengio, Y. Empirical evaluation of gated recurrent neural networks on sequence modeling. In Proceedings of the NIPS 2014 Deep Learning and Representation Learning Workshop, Montreal, QC, Canada, 12 December 2014.
51. Bahdanau, D.; Cho, K.; Bengio, Y. Neural machine translation by jointly learning to align and translate. In Proceedings of the ICLR 2015: International Conference on Learning Representations, San Diego, CA, USA, 7–9 May 2015.
52. Chung, J.; Gulcehre, C.; Cho, K.; Bengio, Y. Gated feedback recurrent neural networks. In Proceedings of the 32nd International Conference on Machine Learning, Lille, France, 6–11 July 2015; pp. 2067–2075.
53. Lyu, H.; Lu, H.; Mou, L. Learning a transferable change rule from a recurrent neural network for land cover change detection. *Remote Sens.* **2016**, *8*, 506. [CrossRef]
54. Lyu, H.; Lu, H.; Mou, L.; Li, W.; Wright, J.; Li, X.; Gong, P. Long-term annual mapping of four cities on different continents by applying a deep information learning method to landsat data. *Remote Sens.* **2018**, *10*, 471. [CrossRef]
55. LeCun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-based learning applied to document recognition. *Proc. IEEE* **1998**, *86*, 2278–2324. [CrossRef]
56. Hubel, D.H.; Wiesel, T.N. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *J. Physiol.* **1962**, *160*, 106–154. [CrossRef]
57. Szeliski, R. Computer vision: Algorithms and applications. In *Springer Science & Business Media*; Springer: London, UK, 2010.
58. Dalal, N.; Triggs, B. Histograms of oriented gradients for human detection. In Proceedings of the International Conference on Computer Vision & Pattern Recognition (CVPR '05), San Diego, CA, USA, 20–26 June 2005; pp. 886–893.
59. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. In Proceedings of the Twenty-Sixth Conference on Neural Information Processing Systems, Lake Tahoe, NV, USA, 3–8 December 2012; pp. 1097–1105.
60. Zeiler, M.D.; Fergus, R. Visualizing and understanding convolutional networks. In *European Conference on Computer Vision*; Springer: Cham, Switzerland, 2014; pp. 818–833.
61. Wan, X.; Zhao, C.; Wang, Y.; Liu, W. Stacked sparse autoencoder in hyperspectral data classification using spectral-spatial, higher order statistics and multifractal spectrum features. *Infrared Phys. Technol.* **2017**, *86*, 77–89. [CrossRef]
62. Mou, L.; Ghamisi, P.; Zhu, X.X. Unsupervised spectral–spatial feature learning via deep residual Conv–Deconv network for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* **2017**, *56*, 391–406. [CrossRef]
63. Kampffmeyer, M.; Salberg, A.B.; Jenssen, R. Semantic segmentation of small objects and modeling of uncertainty in urban remote sensing images using deep convolutional neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 1–9.
64. Maggiori, E.; Tarabalka, Y.; Charpiat, G.; Alliez, P. High-resolution aerial image labeling with convolutional neural networks. *IEEE Trans. Geosci. Remote Sens.* **2017**, *55*, 7092–7103. [CrossRef]
65. Audebert, N.; Le Saux, B.; Lefèvre, S. Beyond RGB: Very high resolution urban remote sensing with multimodal deep networks. I*ISPRS J Photogramm. Remote Sens.* **2018**, *140*, 20–32. [CrossRef]

66. Kussul, N.; Lavreniuk, M.; Skakun, S.; Shelestov, A. Deep learning classification of land cover and crop types using remote sensing data. *IEEE Geosci. Remote Sens. Lett.* **2017**, *14*, 778–782. [CrossRef]
67. Land Use and Coverage Area frame Survey (LUCAS) Details. Available online: https://ec.europa.eu/eurostat/web/lucas (accessed on 14 January 2019).
68. Sentinel-2 MSI Technical Guide. Available online: https://sentinel.esa.int/web/sentinel/technical-guides/sentinel-2-msi (accessed on 14 January 2019).
69. Kaufman, Y.J.; Sendra, C. Algorithm for automatic atmospheric corrections to visible and near-IR satellite imagery. *Int. J. Remote Sens.* **1988**, *9*, 1357–1381. [CrossRef]
70. Mellet, E.; Tzourio, N.; Crivello, F.; Joliot, M.; Denis, M.; Mazoyer, B. Functional anatomy of spatial mental imagery generated from verbal instructions. *Open J. Neurosci.* **1996**, *16*, 6504–6512. [CrossRef]
71. Reddi, S.J.; Kale, S.; Kumar, S. On the convergence of adam and beyond. In Proceedings of the ICLR 2018 Conference Track, Vancouver, BC, Canada, 30 April–3 May 2018.
72. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. In Proceedings of the 3rd International Conference on Learning Representations (ICLR2015), San Diego, CA, USA, 7–9 May 2015.
73. Smith, L.N. Cyclical learning rates for training neural networks. In Proceedings of the 2017 IEEE Winter Conference on Applications of Computer Vision (WACV), Santa Rosa, CA, USA, 24–31 March 2017; pp. 464–472.
74. Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* **2014**, *15*, 1929–1958.
75. Congalton, R.G. A review of assessing the accuracy of classifications of remotely sensed data. *Remote Sens. Environ.* **1991**, *37*, 35–46. [CrossRef]
76. Conrad, C.; Dech, S.; Dubovyk, O.; Fritsch, S.; Klein, D.; Löw, F.; Zeidler, J. Derivation of temporal windows for accurate crop discrimination in heterogeneous croplands of Uzbekistan using multitemporal RapidEye images. *Comput. Electron. Agric.* **2014**, *103*, 63–74. [CrossRef]
77. Skakun, S.; Kussul, N.; Shelestov, A.Y.; Lavreniuk, M.; Kussul, O. Efficiency assessment of multitemporal C-band Radarsat-2 intensity and Landsat-8 surface reflectance satellite imagery for crop classification in Ukraine. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2015**, *9*, 3712–3719. [CrossRef]
78. Rußwurm, M.; Körner, M. Multi-temporal land cover classification with sequential recurrent encoders. *ISPRS Int. J. Geoinf.* **2018**, *7*, 129. [CrossRef]
79. Bergstra, J.; Bengio, Y. Random search for hyper-parameter optimization. *J. Mach. Learn. Res.* **2012**, *13*, 281–305.
80. Vuolo, F.; Neuwirth, M.; Immitzer, M.; Atzberger, C.; Ng, W.T. How much does multi-temporal Sentinel-2 data improve crop type classification? *Int. J. Appl. Earth Obs. Geoinf.* **2018**, *72*, 122–130. [CrossRef]
81. Peña-Barragán, J.M.; Ngugi, M.K.; Plant, R.E.; Six, J. Object-based crop identification using multiple vegetation indices, textural features and crop phenology. *Remote sens. Environ.* **2011**, *115*, 1301–1316. [CrossRef]
82. Fernández-Delgado, M.; Cernadas, E.; Barro, S.; Amorim, D. Do we need hundreds of classifiers to solve real world classification problems? *J. Mach. Learn. Res.* **2014**, *15*, 3133–3181.
83. Shi, D.; Yang, X. An assessment of algorithmic parameters affecting image classification accuracy by random forests. *Photogramm. Eng. Remote Sens.* **2016**, *82*, 407–417. [CrossRef]
84. Lawrence, R.L.; Wood, S.D.; Sheley, R.L. Mapping invasive plants using hyperspectral imagery and Breiman Cutler classifications (RandomForest). *Remote Sens. Environ.* **2006**, *100*, 356–362. [CrossRef]


© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).