The Journal of Supercomputing (2024) 80:1367–1395
https://doi.org/10.1007/s11227-023-05533-4

# Semantic segmentation of satellite images for crop type identification in smallholder farms

Preetpal Kaur Buttar[1] · Manoj Kumar Sachan[1]

Accepted: 30 June 2023 / Published online: 16 July 2023

## Abstract

Accurate and reliable crop type identification from satellite images provides a foundation for crop yield predictions which paves the way to help ensure food security. Most of the work done in the field of crop type mapping using remote sensing is restricted to the developed countries having large field parcels, while a little effort has been directed towards doing so for developing countries, where this task becomes more challenging due to the small size of field parcels, irregular shapes of the fields, and an acute shortage of labelled datasets for training supervised machine learning models. In this research, we try to fill this gap in the literature by exploring the feasibility of performing the semantic segmentation of agricultural fields from satellite images by proposing an encoder–decoder-based semantic segmentation architecture, CropNet, with a ResNet network as the encoder backbone and the use of attention modules in the decoder to allow the model to focus on more important portions of the feature maps and the feature fusion to concatenate the feature maps from all the decoder nodes getting a more precise prediction by bringing the spatial location information from the previous layers. The architecture outperformed the state of the art by 0.51% and 1.3%, on overall accuracy and macro-$F1$ score, respectively, after being trained on the "2019 Zindi's Farm Pin Crop Detection" dataset of Sentinel-2 images. The model achieved a field-wise overall classification accuracy of 78.06% and macro-$F1$ score of 67.3% and a pixel-wise segmentation mean Intersection over Union (mIoU) of 62.22% which is an improvement of 2.56% over the state-of-the-art methods, thereby demonstrating that our model is computationally efficient for the job of semantic segmentation of crop types from the satellite images in the difficult scenario of smallholder farms.

**Keywords** Semantic segmentation · Remote sensing · Encoder–decoder architecture · Sentinel-2 · Crop type mapping · Smallholder farms

✉ Preetpal Kaur Buttar
preetpal@sliet.ac.in

[1] Department of Computer Science and Engineering, Sant Longowal Institute of Engineering and Technology, Longowal, Punjab 148106, India

## 1 Introduction

The 2030 Agenda for Sustainable Development, which comprises 17 Sustainable Development Goals (SDGs), was endorsed by the United Nations in 2015 (https://sdgs.un.org/2030agenda). One of the SDGs concerns food security. Around 795 million people across the world do not have adequate food to eat (https://www.wfp.org/publications/2019-hunger-map). According to an estimate, in the next 35 years, the world will need to produce more food than ever produced in human history due to the factors such as increasing population, climate change, rising incomes, and changing diets. The situation in Africa is even worse with the highest fraction of its population suffering from hunger as compared to any other continent.

An essential task in precision agriculture is the accurate and reliable recognition of crops, paving the way to food security by providing a basis for crop yield forecasting. It can help the government, farmers, buyers, and other stakeholders for making necessary strategies and arrangements for production, harvesting, procurement, stocking, marketing, etc. It can help inform crop diversity, growth patterns, and trends and facilitate crop monitoring. Traditional techniques for crop detection depend on locally collected survey data such as farmer surveys from field visits, weather patterns, rainfall statistics, soil properties, and other elements that affect crop development [1]. These techniques are expensive and difficult to scale. Moreover, precise information may not always be available to make reliable predictions. Satellite images serve as a source of an enormous amount of data about land use, which can be explored to get very useful insights into crop growth patterns over vast geographical extents. Satellite data are multispectral, consisting of optical (visible and near infrared (NIR)), thermal, and microwave bands. Easily accessible satellite images provide large spatial coverage as well as are very helpful for collecting information in close to real time at the regional scale since they have a high temporal revisit frequency [2]. At present, more than 4,500 satellites are orbiting the earth with more than 600 being imagery satellites. Current state-of-the-art satellites have a resolution of 25 cm per pixel, which means that a person takes 3 pixels on an image.

Semantic segmentation of crops implies the process of crop identification and accurate localization of crops in the image at the pixel level [3, 4]. The goal is to assign each pixel in an image a class label to symbolize the crop to which that pixel belongs employing semantic segmentation. Several studies have revealed that satellite images can be used to predict the area where each type of crop is planted. The approaches to crop type detection from satellite images can be broadly classified into traditional approaches based on feature handcrafting and machine learning approaches including deep learning. Earlier approaches to crop detection using remote sensing data relied on handcrafted features which used various types of vegetation indices such as the Normalized Difference Vegetation Index (NDVI) [5, 6], Enhanced Vegetation Index [7, 8], and Normalized Difference Water Index (NDWI) [7, 9]. The drawback of this strategy is that the vegetation indices employed are extremely rudimentary indices that only contain data from a few (often two) of the available spectral bands [1].

Machine learning techniques applied to multispectral satellite images include support vector machines [8, 10–17], decision trees [6, 10, 18, 19], random forest [18, 20, 21], ridge regression [7], multilayer perceptron [10], restricted Boltzmann machine [8], and maximum likelihood classification [15]. These approaches require manual feature extraction, and the classification accuracy of these approaches is limited by the representation ability of the manually extracted features. Another major shortcoming of these approaches for crop type classification is that they perform poorly outside their region of interest [22].

The fields of computer vision and natural language processing have advanced more swiftly recently because of the tremendous growth in deep learning research during the past several years. In the image classification challenge on the ImageNet dataset, deep learning-based convolutional neural networks (CNNs) have been able to do nearly as well as humans [23]. Other image-based operations, like object detection, image segmentation, image synthesis, etc., have also seen notable advancements. The application of deep learning is penetrating newer areas including remote sensing. Applying deep learning techniques to the satellite data can be helpful in semantically segmenting different crops. Pertinent features from the multispectral images can be extracted using deep learning techniques automatically in an end-to-end manner, completely eliminating the feature engineering tasks. A combination of an unsupervised neural network and an ensemble of supervised neural networks was proposed as a multilevel deep learning architecture by Kussul et al. [24] for classifying crop types and land cover. Zhong et al. [25] built two different types of deep learning models: one based on 1D convolutional layers and the other on long short-term memory (LSTM). Cai et al. [26] also used 1D CNN for crop type mapping. Ji et al. [27] proposed a 3D CNN for automatically classifying crops from multitemporal satellite images. Terliksiz and Altýlar [28] used spatio-temporal features with a 3D CNN model to predict soybean yield. In their study, Wang et al. [29] employed crowdsourcing to gather ground truth crop type data and used CNNs for mapping rice and cotton crops in southeast India at 10 m resolution. Rußwurm et al. [22] developed the BreizhCrops dataset for mapping crop types in the Brittany region of France. Seven classification algorithms were also used, including a random forest classifier and six deep learning techniques based on convolution, recurrence, or attention. RNN was employed in [30, 31] for mapping various crop types across time. In [32], ResUNet-a, a deep convolutional neural network with a fully connected U-Net backbone, was used to approach the issue of retrieving field borders from the satellite images. Rustowicz et al. [33] used two approaches to the issue of crop type semantic segmentation in Africa: one used a two-dimensional (2D) U-Net and convolutional LSTM, and the other used a 3D CNN. The limitation of RNN and LSTM's ability to compute many layers simultaneously can drastically lengthen computation times and make it impractical for large-scale crop mapping. The sparse swapping and parameter sharing features of CNN may be able to shorten the time required for network training [34].

Another issue prevalent in precision agriculture which can help increase the agricultural productivity and enhance food security is that of crop pest and disease control. Manual identification of various kinds of pests and crop diseases is problematic due to the lack of expertise among the agricultural workers and being labour intensive. Therefore, their automatic detection through computer vision techniques has a major role to play. Various researchers have approached this problem through advanced deep learning-based mechanisms. For instance, Jiao et al. [35] proposed an anchor-free CNN for detection and classification of 24 types of pests, Dong et al. [36] proposed a channel recalibration feature pyramid architecture for the detection of small pests, Jiao et al. [37] proposed adaptive feature fusion for feature pyramid network to detect multiple classes of pests in complex scenarios, Li et al. [38] developed a multibranch CNN with density mapping for aphid counting, and Dong et al. [39] proposed a multiscale feature pyramid network and an adaptive feature region proposal network for the detection of multiple pest categories.

Most of the literature on crop type mapping using satellite images is concentrated on large commercial agricultural fields in developed countries such as [21, 24–26, 28, 31], etc. These farms are characterized by large size (5 to 10 hectares on average), definite shape, and a single crop throughout the season as opposed to smallholder farms with size lesser than 5 hectares [40], irregular shapes, diverse intercropping, and loosely defined boundaries. Large farms yield reasonable size and number of satellite images to be fed to a CNN-based feature extraction model giving sufficient accuracy [41]. Crop diversity has a little impact on the spectral band readings for a given field because the same crops are present in the nearby fields. However, about 30% of the world's food supply is produced through smallholder farming, which uses 12% of the world's agricultural area [42, 43]. In Africa, the importance of these smallholder farms is even more where around 60% of the labour market is employed by smallholder farmers, who cultivate more than 80% of the cropland [42]. Due to the location-specific nature of crop phenotypic and phenology information as well as the variations in canopy-level spectral reflectance among various environments and management practises, crop type detection models calibrated for one region cannot be easily generalized to another region [44]. The models trained on large commercial farms cannot be effectively applied on smallholder farms as in smallholder farms, crop diversification is highly prevalent, fields boundaries are blurry, a single image patchlet contains multiple crop classes but only a few representative pixels for a given crop because of small field size, and the impact of nearby crops is substantial [45, 46]. Thus, the automatic detection of crops sown in smallholder farms is still a challenging task [33, 41, 46–49].

In this work, we apply deep learning-based convolutional encoder–decoder architecture on Sentinel-2 satellite images to detect and semantically segment the crops grown in different fields in the difficult scenario of smallholder farms in the African continent. The size of the fields in our work is comparatively small compared to the size of the fields in industrial farming settings, which poses a significant challenge. The small size of fields means a lesser number of pixels in the image carrying useful discriminative information. The field of deep learning and computer vision is constantly evolving, with new architectures and techniques being developed regularly. CNN-based encoder–decoder segmentation architectures became the mainstream with the introduction of fully convolutional networks (FCNs) [50]. First proposed for the Natural Language Processing domain [51], the self-attention-based transformer architectures found their utility for computer vision tasks including semantic segmentation [52–55]. In terms of segmentation accuracy, transformer-based models outperformed CNN-based and other models on a variety of benchmarks [56]. Transformer is a suitable option for segmentation architecture design if the objective is to increase network accuracy without considering model size and computational expense. Although transformer-based techniques match state of the art in terms of accuracy, the CNN-based models continue to make up the majority of segmentation models having undeniable advantages in terms of network size and inference speed. The reason behind this is that with only the parameters of the fixed-size window to learn and no need to encode global information, the convolution operation extracts the image features through a fixed-size convolution kernel. However, convolution has the drawback of being unable to detect long-distance relationships, such as those between arbitrary pixels in an image. To capture global information, we incorporated non-local self-attention mechanism into our proposed architecture to leverage the benefits of transformer-based attention into CNN-based architecture while keeping the computational cost low. Specifically, we employed lightweight SE attention modules after the convolution operations to amplify the useful features and suppress the irrelevant ones. The encoder backbone used in the architecture is that of pretrained ResNet model, which helps the training to advance quickly as its weights are already tuned to detect the low-level features such as edges, lines, and curves. The use of feature fusion bars to concatenate the decoder layer feature maps helps in getting a more precise prediction by bringing the spatial location information from the previous layers. All these design choices helped us to achieve high segmentation scores even on a small dataset. Our work has contributed in the following ways:

- A robust crop type segmentation model was proposed for the difficult scenario of smallholder farms with irregular shapes, diverse inter-cropping, and loosely defined boundaries using a multicrop, multispectral, and multitemporal dataset.
- A working pipeline to automate the workflow for preprocessing multitemporal satellite images for various tasks including cloud removal, adding spectral indices to the images, and linear interpolation through time was proposed.
- The performance of different types of attention modules for segmenting crop types using the proposed model was compared.
- The effects of using different combinations of satellite bands and spectral indices on the performance of crop type segmentation by the proposed model were studied.
- The effects of the proposed design choices, i.e. attention modules and feature fusion bars, on the performance of the proposed model were studied.

The remainder of this article is organized as follows. Section 2 presents the dataset and explains the various preprocessing tasks performed on the dataset as well as a detailed description of the proposed architecture. Section 4 puts forth the experimental setup with implementation details along with the findings and discussion. Section 5 wraps up the work and considers its future directions.

# 2 Data and Methods

## 2.1 Data and its preprocessing

The dataset used for our experiments is ‘Farm Pin Crop Detection Challenge’ (https://zindi.africa/competitions/farm-pin-crop-detection-challenge) at Zindi, which is a competitive data science platform in Africa that focuses on data science for social good. The ‘Farm Pin Crop Detection Challenge’ is originally a classification task that seeks to categorize crops in South African fields along a stretch of the Orange River, which is our Area of Interest (AoI). This dataset was previously used in the work of [47], wherein they compared the classical machine learning approaches including K-nearest neighbours, random forest, and gradient boosting with a deep learning-based architecture, U-Net. The authors achieved the best results using gradient boosting with an overall accuracy of 77.4% and a macro-$F1$ score of 66%. The dataset consists of Sentinel-2 satellite images for 11 points in time between January and August 2017 over AoI covering the summer and winter months. The dataset consists of 2497 fields with nine crop labels, namely cotton, dates, grass, lucerne, maize, pecan, vineyard, vineyard and pecan (intercrop), and vacant (for vacant fields), as shown in Table 1. Additionally, there is also a background class label. The field boundaries are represented as polygons in a shapefile.

This dataset is limited by its small size with less than 3000 fields. Moreover, the size of the fields is also small. This poses a typical challenge to automatic crop type detection. However, the dataset is diverse with nine different crop classes. The classes such as cotton, lucerne, maize, pecan, vacant, and vineyard are represented well with enough number of fields, but the classes such as dates and intercrop are under-represented.

Three out of the total number of fields in the dataset contain NaN values, i.e. not a number in one of their attributes; thus, we dropped those fields and were left with 2494 fields. Then, we dropped the fields with ‘dates’ as there are only 7 field samples representing this crop in the dataset, which is not sufficient enough to train a deep learning-based model. We also dropped ‘intercrop’ which represents a mixed crop of ‘vineyard’ and ‘pecan’, already present in the dataset. By doing this, we were left with 2410 fields in the dataset, which we divided into training field sets and test field sets in the ratio of 75–15%, leaving 2050 and 360 fields in the training field set and test field set, respectively. The fields in the test field set were randomly sampled to include a sufficient number of examples of each class of crops.

The presence of some relatively small fields as compared to those in the datasets of industrial farming in other countries such as the USA is a challenge to the semantic segmentation of crops in the dataset.

We employed the eo-learn library for the preprocessing of the Sentinel-2 image dataset. It is an open-source Python library that automates the tasks involved in processing satellite images, starting with downloading Sentinel-1A and Sentinel-2A images and continuing with feature extraction, preprocessing, and applying

**Table 1** Number of fields per crop in the dataset

| | Cotton | Dates | Grass | Lucern | Maize | Pecan | Vacant | Vineyard | Intercrop | Total |
|---|---|---|---|---|---|---|---|---|---|---|
| Dataset before processing | 142 | 7 | 95 | 536 | 283 | 147 | 267 | 943 | 77 | 2497 |
| Dataset after removing NaNs | 142 | 7 | 95 | 536 | 283 | 146 | 267 | 941 | 77 | 2494 |
| *Dataset after removing dates and intercrop classes* | | | | | | | | | | |
| Total fields | 142 | 0 | 95 | 536 | 283 | 146 | 267 | 941 | 0 | 2410 |
| Training field set | 120 | 0 | 74 | 453 | 239 | 125 | 226 | 813 | 0 | 2050 |
| Test field set | 22 | 0 | 21 | 83 | 44 | 21 | 41 | 128 | 0 | 360 |

machine learning models to those images (https://github.com/sentinel-hub/eo-learn). We were able to break our AoI into patches, construct a workflow, and then run the workflow on many patches in parallel using eo-learn. As shown in Fig. 1, the workflow consists of loading the images from the disc, removing too cloudy scenes, adding NDVI, NDWI, and Euclidean norm features to the images, adding a raster mask with the target crop for each field polygon, generating different datasets by combining different spectral bands with the spectral indices, resampling through time, and saving the resulting data in the form of EOPatches, a format for storing geospatial data by eo-learn.

The various preprocessing steps are explained as under:

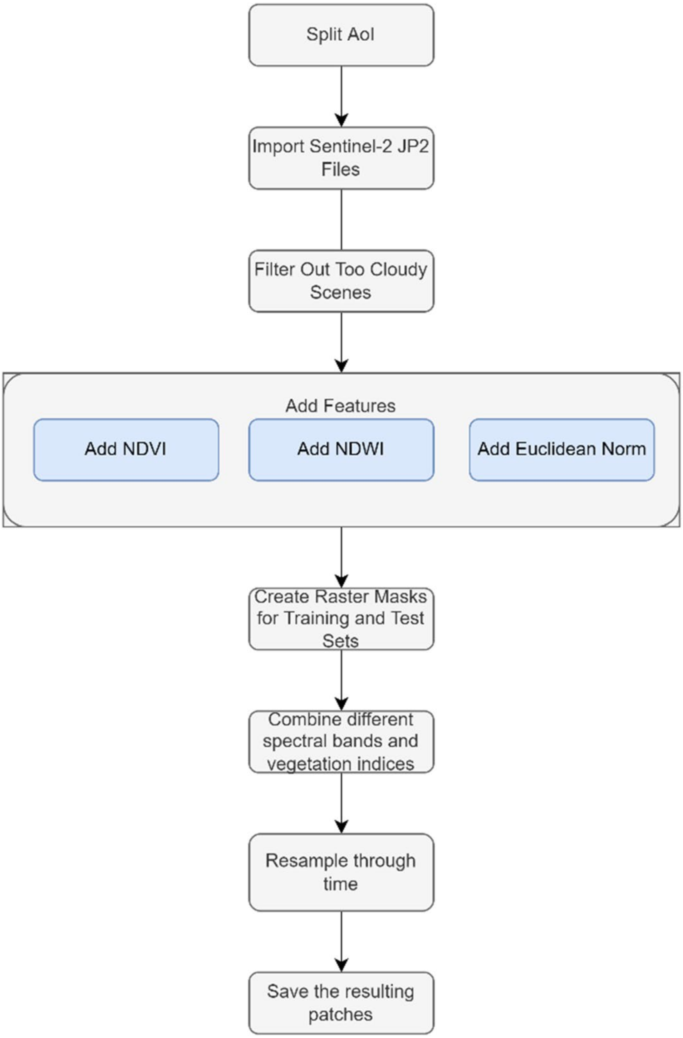

Split AoI
Import Sentinel-2 JP2 Files
Filter Out Too Cloudy Scenes
Add Features
Add NDVI
Add NDWI
Add Euclidean Norm
Create Raster Masks for Training and Test Sets
Combine different spectral bands and vegetation indices
Resample through time
Save the resulting patches

**Fig. 1** The eo-learn workflow

### 2.1.1 Splitting AoI

We used the BBoxSplitter function from the Sentinel Hub library for splitting the AoI into 12 patches. Given the area, it calculates the bounding box and splits it into smaller bounding boxes of equal sizes. Then it filters out the bounding boxes that do not intersect the area. The resulting patches are eo-learn EOPatches, where an EOPatch is a common data structure that contains multitemporal remote sensing data of a single patch typically defined by a bounding box in a specific coordinate system. It can also hold extracted features such as NDVI, Euclidean norm, etc., metadata, and corresponding vector data. The result of this step is shown in Fig. 2.

### 2.1.2 Loading image data from disc

The image data in the collection are provided in JPEG2000 format, in a typical Sentinel-2 folder structure, and it has not been resampled or scaled to an AoI. It was stacked date-wise and band-wise as a four-dimensional NumPy array of shape $date \times width \times height \times band$ ($11 \times 1345 \times 1329 \times 13$) in their respective patches.

### 2.1.3 Cloud masking

Then, we employed s2cloudless model and the SSIM-based multitemporal classifier (https://medium.com/sentinel-hub/on-cloud-detection-with-multi-temporal-data-f64f9b8d59e5) for cloud masking and retained the scenes which contained more than 80% valid pixels and discarded the others.

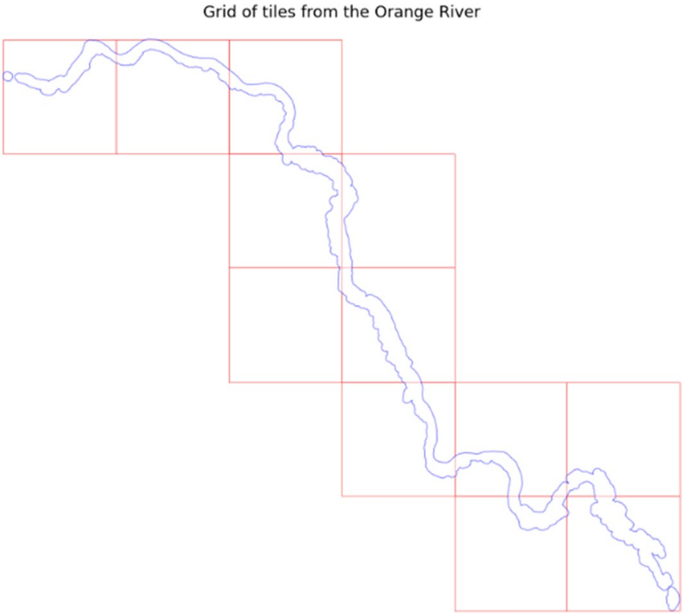


**Fig. 2** Resulting EOPatches after splitting AoI into 12 patches

### 2.1.4 Feature extraction

We then extracted the spectral indices (https://www.indexdatabase.de/db/i.php) NDVI, NDWI, and the Euclidean norm. NDVI is used to quantify vegetation health. It is calculated based on the concept that leaves absorb a high amount of visible red light and reflects a high amount of NIR. NDVI is calculated as:

$$\mathrm{NDVI} = \frac{\mathrm{NIR} - \mathrm{Red}}{\mathrm{NIR} + \mathrm{Red}} \tag{1}$$

The value of NDVI ranges between $-1$ and $+1$ with a higher NDVI value indicating a healthier plant. As we are dealing with the task of semantic segmentation of crops, extracting NDVI serves as a useful feature for identifying crops as well as distinguishing among different crops. Different crops are planted at different times, grow at different rates, and are harvested at different times. As a result, NDVI changes differently over time for different crops [57]. We created an EOTask to calculate NDVI between two bands.

NDWI [58] is an index to detect changes in the water content of leaves, which is calculated as:

$$\mathrm{NDWI} = \frac{\mathrm{Green} - \mathrm{NIR}}{\mathrm{Green} + \mathrm{NIR}} \tag{2}$$

NDWI varies from $-1$ to $+1$, depending on the vegetation type and cover as well as the water content of the leaves. High vegetation water content and vegetation fraction cover are correlated with high NDWI readings, while low vegetation water content and low vegetation cover are correlated with low NDWI values.

We also calculated the Euclidean norm of all bands within an image as:

$$\mathrm{Norm} = \sqrt[2]{\sum_i {B_i}^2} \tag{3}$$

where $B_i$'s are the individual bands within an image.

### 2.1.5 Adding target masks

We generated target masks for the training as well as test field set images to treat the crop classification problem as a semantic segmentation task, where crop types are indicated by the raster layer. We availed the benefit of the knowledge in the local spatial context of each field by rephrasing the crop identification problem as a semantic segmentation challenge. It also allowed for more training data to be generated through repeated sampling. Another raster mask layer indicating just the field polygons was also generated to be used for inference at a later stage. Figure 3 shows (a) a satellite image in the visible spectrum over one of the patches in AoI, (b) its corresponding raster mask of crop types in each field polygon for the training field set, and (c) the corresponding raster mask of crop types in each field polygon for the test field set.

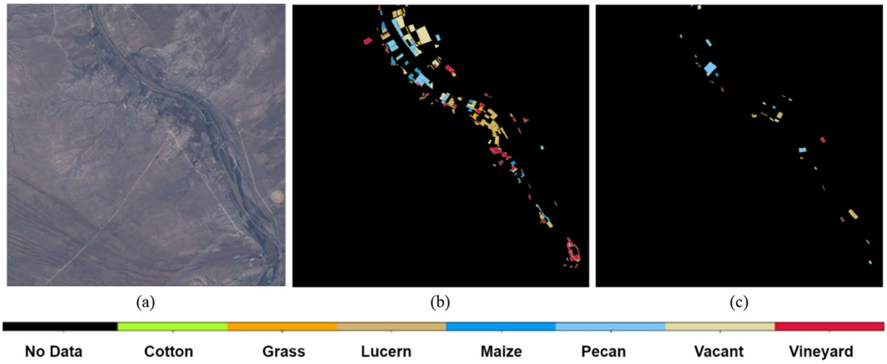


**Fig. 3** **a** A satellite image in the visible spectrum over one of the patches in AoI, **b** its corresponding raster mask of crop types in each field polygon for the training field set, and **c** the corresponding raster mask of crop types in each field polygon for the test field set

### 2.1.6 Generating experimental datasets

We created four different datasets to experiment with based on the different combinations of spectral bands and spectral indices. Table 2 provides the specifications of these datasets. One dataset contains the 13 Sentinel-2 spectral bands only, while the other three datasets included the three spectral indices: NDVI, NDWI, and Euclidean norm in addition to the spectral bands. They differ in the number of Sentinel-2 spectral bands included.

### 2.1.7 Linear interpolation

Cutting out clouds leaves gaps in the data for the areas with cloud cover in each time slice. These gaps can be filled by interpolating between preceding and subsequent time slices. For this, we used linear interpolation to average out data to approximately 1 time slice per month, which resulted in reducing the time dimension from 11 time points to 8 time points between January 01 and August 20, 2017, the time frame during which the Sentinel-2 images for the dataset were captured.

**Table 2** Different datasets used for experimentation based on different combinations of spectral bands and spectral indices

| Dataset name | Spectral bands and spectral indices included | Total number of channels |
|---|---|---|
| 7_channels | Red, Green, Blue, NIR, NDVI, NDWI, Euclidean Norm | 7 |
| 9_channels | Red, Green, Blue, NIR, SWIR band-11, SWIR band-12, NDVI, NDWI, Euclidean Norm | 9 |
| 13_channels | All 13 Sentinel-2 bands | 13 |
| 16_channels | All 13 Sentinel-2 bands, NDVI, NDWI, Euclidean Norm | 16 |

Additionally, to deal with any gaps at the start or end of the time period, we employed a simple extrapolation technique to copy values from preceding or succeeding time points, as necessary.

## 2.2 The proposed architecture

The proposed architecture's base is a deep convolutional encoder–decoder architecture, which is a kind of FCN [50]. An FCN takes an input image and produces its corresponding mask image indicating the predicted labels for each pixel. All the layers in an FCN are convolutional layers and there are no fully connected dense layers unlike a normal convolutional network which contains fully connected layers at the tail of the network, typically used for image classification task. Our proposed architecture, CropNet, is based on an FCN called U-Net++ [59] with a ResNet-152-based encoder and the decoder consisting of attention modules and the feature fusion bars technique. The convolutional layers in the proposed architecture are arranged in an encoder–decoder structure. The encoder's job is to gradually downscale the spatial resolution of the input image using pooling layers while extracting the relevant feature maps at each layer using convolutions. The decoder's purpose is to capture the specifics of the segmentation using the learnt features and gradually increase the spatial resolution at each layer by combining the feature maps generated at the previous decoder layer and the corresponding encoder layer through skip connections. The last decoder layer, in this manner, generates the final feature map, which is the semantic segmentation map of the input image. CropNet consists of three major improvements as compared to the baseline U-Net++ architecture:

(1) Use of ResNet architecture as the encoder
(2) Addition of an attention module at each decoder layer
(3) Use of feature fusion bars

### 2.2.1 Use of a ResNet variant as the encoder

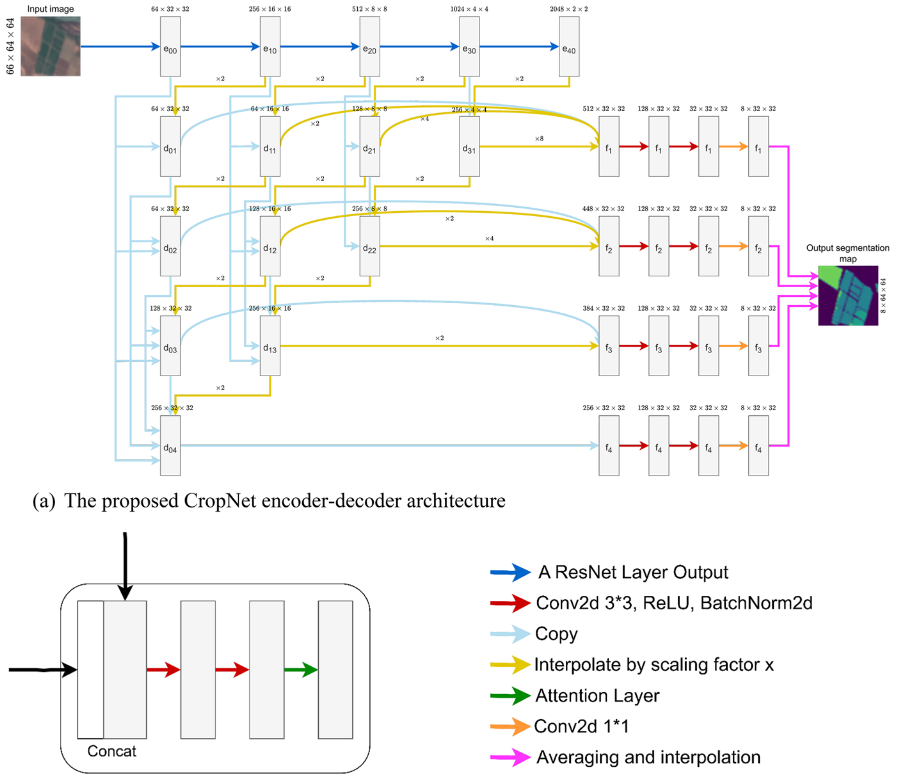
Instead of using the original encoder of the U-Net++ model, the proposed architecture consists of a U-Net++ network with a ResNet acting as the encoder. ResNet [60] is a popular CNN architecture for image classification. We adopted ResNet-152 as the encoder backbone in our experiments. The reason for using a ResNet encoder lies in its ability to train deeper neural networks. When neural networks are sufficiently deep, they develop issues like vanishing gradients, the curse of dimensionality, and the degradation problem, which causes accuracy to stop improving beyond a certain point and finally start declining. ResNet's residual networks are built from residual blocks with skip connections, allowing the layers to learn the residual between the input and the output rather than attempting to learn the true input. Either the layers in residual blocks are trained or skip connections are used to forego their training.

Accordingly, based on how the error travels backward in the network, different regions of networks will be trained at different rates for different training data points. This can be compared to training an ensemble of various models on a dataset to achieve the highest accuracy. Larger gradients are transmitted to the first layers via skip connections, allowing for faster learning comparable to that of the final layers. Deeper networks can be trained as a result.

The ResNet-based encoder for CropNet extracts the feature maps from the deepest layer at each stride. The input to the encoder is of size $m \times 64 \times 64$, where $m$ is the number of channels in the input image, and the image is 64 pixels wide and high. Figure 4 shows the dimensions of the feature maps extracted at each encoder layer, $e_{00}$, $e_{10}$, $e_{20}$, $e_{30}$, and $e_{40}$.

### 2.2.2 CropNet decoder with attention

Each layer of CropNet decoder is composed of several decoder nodes, where each decoder node consists of a concatenation of the input feature maps, then a sequence of two convolutional layers and an attention layer (see Fig. 4b). Each convolutional layer in turn is a sequence of a 2D convolution with kernel size



(a) The proposed CropNet encoder-decoder architecture

(b) The operations performed inside a decoder node

**Fig. 4** The proposed architecture

$3 \times 3$, a rectified linear unit (ReLU) activation, and 2D batch normalization, shown by the 'red' arrows in Fig. 4b.

Attention modules at each decoder node are indicated by 'green' arrows in Fig. 4b. Attention in neural networks allows to focus on important features and helps to improve the representations of interests. We employed SE attention [61] in our attention modules. SE is a lightweight channel-wise attention module that enables a network to undergo dynamic channel-wise feature recalibration to improve its representational power. Figure 5 depicts the detailed architecture of a SE attention module. In this case, 'S' denotes the squeeze operation, which conducts a channel-wise 'global average pooling' over the whole feature map, and 'E' is the excitation operation, which conducts the activation using a few fully connected layers and an activation function. The squeeze operation ($F_{sq}$) pools the channel-wise global average over the spatial dimensions ($H \times W$) of a feature map $U \in \mathbb{R}^{H \times W \times C}$ to create the output $S \in \mathbb{R}^{1 \times 1 \times c}$. Specifically, for each channel $c$, $F_{sq}(U_c)$ is calculated as:

$$F_{sq}(U_c) = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} U_c(i,j) \tag{4}$$

The output of the squeeze operation is then fed through the excitation operation ($F_{ex}$) to produce channel-wise modulation weights, $E \in \mathbb{R}^{1 \times 1 \times c}$. $F_{ex}$ is a sequence of a fully connected (FC) layer, ReLU activation, another FC layer, and a sigmoid function. Equation 5 shows the sequence of operations employed:

$$F_{ex}(S, W) = \sigma(W_2 \delta(W_1 z)) \tag{5}$$

where $W_1 \in \mathbb{R}^{c \times \frac{c}{R}}$ and $W_2 \in \mathbb{R}^{\frac{c}{R} \times c}$ are the weight matrices, $\delta$ refers to the ReLU activation, $\sigma$ is the sigmoid function, and $R$ is the reduction ratio (16 by default).

$E$ is applied directly to $U$ through a simple broadcasted element-wise multiplication ($F_{scale}$) which scales each channel in $U$ to its corresponding learned weights in $E$ to produce attention-based feature maps that will be supplied to later layers of the neural network.

The decoder nodes are present at four layers, $l = 0, 1, 2, 3$. The nodes present at the first layer, $l = 0$, are $d_{01}$, $d_{02}$, $d_{03}$, and $d_{04}$. At layer $l = 1$, the nodes $d_{11}$, $d_{12}$, and $d_{13}$ are present. Similarly, the nodes $d_{21}$ and $d_{22}$ make up the layer $l = 2$. The node present at $l = 3$ is $d_{31}$.

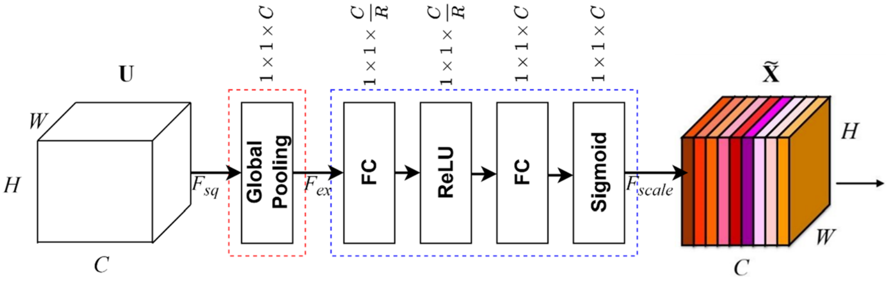


**Fig. 5** An SE attention module [61]

The nodes in the decoder are connected through skip connections. Each decoder node gets as input the feature maps from the corresponding encoder layer, and from all the previous decoder nodes present in the node's corresponding layer, as well as the upsampled output from the previous encoder/decoder node at the lower layer.

At decoder node $d_{i1}$, $0 \leq i \leq 3$, an output feature map of size $64 \times 32 \times 32$ is produced through a sequence of operations given by Eq. 6:

$$d_{i1} = se(conv_2(conv_1(e_{i0} + upsample(e_{(i+1)0})))), 0 \leq i \leq 3 \tag{6}$$

where *upsample* operation is meant to double the spatial resolution of the input feature map, + refers to the concatenation of the corresponding feature maps along the channel dimension, $conv_1$ is the first convolution operation, and $conv_2$ is the second convolution operation. $conv_1$ reduces the number of channels to 32, 32, 64, and 128, and $conv_2$ to 64, 64, 128, and 256, respectively, in $d_{01}$, $d_{11}$, $d_{21}$ and $d_{31}$. The spatial resolution of the feature map at any decoder node in a layer remains the same as that of corresponding encoder node in that layer. In a similar manner, feature maps at decoder nodes $d_{02}$, $d_{12}$, and $d_{22}$ are generated as given by Eq. 7:

$$d_{i2} = se(conv_2(conv_1(e_{i0} + d_{i1} + upsample(d_{(i+1)1})))), 0 \leq i \leq 2 \tag{7}$$

Here, $conv_1$ reduces the number of channels to 32, 64, and 128, and $conv_2$ to 64, 128, and 256, respectively, in $d_{02}$, $d_{12}$, and $d_{22}$. In the next step, the feature maps of the decoder nodes $d_{03}$ and $d_{13}$ were computed as given by Eq. 8:

$$d_{i3} = se(conv_2(conv_1(e_{i0} + d_{i1} + d_{i2} + upsample(d_{(i+1)2})))), 0 \leq i \leq 1 \tag{8}$$

Here, $conv_1$ reduces the number of channels to 64 and 128, and $conv_2$ to 128 and 256, respectively in $d_{03}$, and $d_{13}$. Lastly, the feature map for node $d_{04}$ of size $256 \times 32 \times 32$ is generated as given by Eq. 9:

$$d_{04} = se(conv_2(conv_1(e_{00} + d_{01} + d_{02} + d_{03} + upsample(d_{13})))) \tag{9}$$

Here, $conv_1$ reduces the number of channels to 128 and $conv_2$ to 256.

### 2.2.3 Use of feature fusion bars

We employed the feature fusion bars technique which simply concatenates or fuses together the feature maps produced by several convolutional layers in the network. This helps in getting a more precise prediction by bringing the spatial location information from the previous layers. In the proposed architecture, we employed the feature fusion bars to concatenate the feature maps from the decoder nodes of CropNet along the channel dimension. Four feature fusion bars $f_1$, $f_2$, $f_3$, and $f_4$ were generated as given by Eqs. 10–13:

$$f_1 = d_{01} + d_{11} + d_{21} + d_{31} \tag{10}$$

$$f_2 = d_{02} + d_{12} + d_{22} \tag{11}$$

$$f_3 = d_{03} + d_{13} \tag{12}$$

$$f_4 = d_{04} \tag{13}$$

For generating each feature fusion bar, the spatial dimensions of the constituent feature maps were first upsampled to be of the same size, i.e., 32 × 32. For instance, to generate $f_1$, the feature maps $d_{11} \in \mathbb{R}^{64\times16\times16}$, $d_{21} \in \mathbb{R}^{128\times8\times8}$, and $d_{31} \in \mathbb{R}^{256\times4\times4}$ were upsampled to become $d_{11} \in \mathbb{R}^{64\times32\times32}$, $d_{21} \in \mathbb{R}^{128\times32\times32}$, and $d_{31} \in \mathbb{R}^{256\times32\times32}$, respectively, while $d_{01} \in \mathbb{R}^{64\times32\times32}$ already has the compatible spatial dimensions. The upsampled feature maps were then concatenated along the channel dimension to generate a feature fusion bar. Four feature fusion bars thus generated were each passed through a sequence of two convolutional layers and a 2D convolution with kernel size 1 × 1 to produce 4 different segmentation maps, each of size 8 × 32 × 32, where 8 is the number of classes in the dataset including a no-data class. These four segmentation maps were then combined into a single final segmentation map of size 8 × 64 × 64.

### 2.3 Implementation details

The proposed architecture for crop segmentation was implemented in Python (version 3.7.13) and the open-source neural network libraries PyTorch (version 1.6.0+cu101) and Fastai (version 1.0.61) on Google Colaboratory cloud platform using Tesla P100 16 GB GPU.

We randomly sampled patchlets of size 64 × 64 pixels from 12 patches. This was done for the training field images as well as the test field images of each of the 4 datasets, as explained in Table 2. Because the fields are small and the available Sentinel-2 imagery has a maximum spatial resolution of 10 m, we kept the patchlet size small. This means a 1 hectare (10,000 m$^2$) field looks in the imagery as 32 × 32 pixels. The patchlets were sampled in such a way that each patchlet contained at least a portion of the training field. We saved two pickle files for each patchlet: one with the input imagery and the other with the crop types as a raster layer. Figure 6 illustrates the NDVI and visible images for nine randomly sampled 64×64 patchlets at a single time point, along with the appropriate target crop types.

For each of the 4 datasets generated, we stacked the total channels $n$ in the dataset with each of the 8 timepoints to produce a $n$ × 8-channel image to get a rank 3 tensor. It resulted in 56 × 64 × 64, 72 × 64 × 64, and 128 × 64 × 64 images, for 7_channels, 9_channels, and 16_channels datasets, respectively. For all the four datasets, 10% of the training field images were kept for validation.

Because the network architecture had many parameters and the dataset was rather limited, we employed data augmentation to prevent overfitting. The various augmentation techniques applied were vertical/horizontal flips, rotation, zoom, warping, and cutout.

The chosen dataset set has an uneven distribution of different crop varieties. Originally introduced in the RetinaNet [62] paper, we employed the focal loss, which is useful in cases with extreme class imbalance as ours. Using the weight parameter

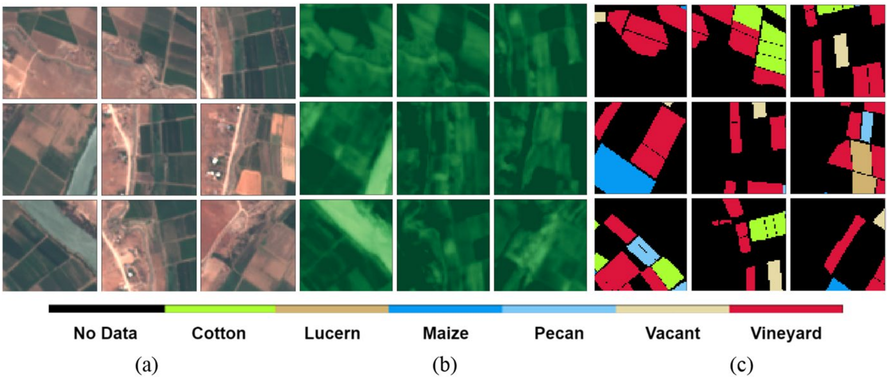


**Fig. 6** **a** Visible spectrum, **b** NDVI images, and **c** the corresponding target crop types at a single time point for nine randomly sampled 64 × 64 training patchlets

of the loss function, we weighed the loss function in proportion to the inverse frequency of each crop type. Much of the training image area lacked a crop type, either because there was no field in the area or because the field was not part of the training dataset, so we ignored predictions where there was no crop type label.

Other hyperparameters used in training the proposed architecture are listed in Table 3.

The model was trained on the architecture of CropNet for all the three datasets, i.e. 7_channels, 9_channels, and 16_channels datasets. For training, we employed a one-cycle training policy with a maximum learning rate of 0.001 for the first 5 epochs, keeping the pretrained encoder weights frozen. We trained the model for a further 15 epochs with a maximum learning rate of 0.0001, allowing the encoder weights to be updated.

## 2.4 Evaluation metrics

Several typical statistical metrics commonly used in the state-of-the-art image segmentation architectures were selected to evaluate the results, which are IoU score, accuracy, precision, recall, and $F1$ score. The value of all these evaluation metrics

**Table 3** Various network hyperparameters used for training the proposed architecture

| Hyperparameter | Value |
|---|---|
| Optimizer | Adam |
| Maximum learning rate | 0.001–0.0001 |
| Regularization techniques | Weight decay |
| Loss function | Focal loss |
| Epochs | 5 + 15 |
| Batch size | 32 |

ranges between 0 and 1. The larger the value of an evaluation metric, the better the segmentation results.

We calculated three accuracy statistics: per-class accuracy, average accuracy, and overall accuracy. Per-class accuracy was calculated for all seven classes in our dataset. For a given class $c$, it is defined as:

$$\text{Accuracy}_c = \frac{tp_c + tn_c}{tp_c + tn_c + fp_c + fn_c}, \tag{14}$$

where $t_p$, $t_n$, $f_p$, and $f_n$ are, respectively, the numbers of true positive (TP), true negative (TN), false positive (FP), and false negative (FN) pixels for the class $c$. Average accuracy is the average of the per-class accuracies of each class. It is defined as:

$$\text{Averageaccuracy} = \frac{\sum_{i=1}^{C} \text{Accuracy}_i}{C}, \tag{15}$$

Here, $C$ is the total number of classes in the dataset. Overall accuracy is simply the total number of correct predictions over all the classes divided by the total number of pixels in the entire test set.

The industry-standard statistic for semantic segmentation is the IoU score, often known as the Jaccard index [63]. It reflects the overlap between the anticipated segmentation mask and the ground truth mask, divided by the total number of pixels in both masks. IoU score for a class c is defined as:

$$IoU_c = \frac{tp_c}{tp_c + fp_c + fn_c}, \tag{16}$$

As we are dealing with multiple classes segmentation, we also compute mIoU as:

$$mIoU = \frac{\sum_{i=1}^{C} tp_i}{\sum_{i=1}^{C} (tp_i + fp_i + fn_i)}, \tag{17}$$

where $C$ is the total number of classes. To reveal more insights into the model's performance, three additional metrics—precision, recall, and $F1$ score—are also used. For a single class c, precision is defined as:

$$\text{Precision}_c = \frac{tp_c}{tp_c + fp_c}, \tag{18}$$

When dealing with multiple classes as in our case, the macro-precision can be calculated as simply the average of the precision values over all the classes in the dataset. It is defined as:

$$Macroprecision = \frac{\sum_{i=1}^{C} Precision_i}{C}, \tag{19}$$

For a single class c, recall is defined as:

$$Recall_c = \frac{tp_c}{tp_c + fn_c}, \tag{20}$$

Similar to average precision, we calculated macro-recall as:

$$Macrorecall = \frac{\sum_{i=1}^{C} Recall_i}{C}, \tag{21}$$

The $F1$ score simply combines both precision and recall as:

$$F1_c = 2 \times \frac{Prec_c \times Recall_c}{Prec_c + Recall_c}, \tag{22}$$

A model with a high $F1$ score demonstrates high precision and recall. In multiclass problems, the macro-$F1$ score is simply the average of the $F1$ scores on individual classes:

$$MacroF1 = \frac{\sum_{i=1}^{C} F1_i}{C}, \tag{23}$$

These metrics provide meaningful insights into understanding the efficacy and performance of the model.

# 3 Results and discussion

## 3.1 Performance comparison among different types of attention modules in terms of the segmentation results on 16_channels dataset

We compared the segmentation performance of four different attention modules before arriving at the decision to use the SE attention module. Apart from SE attention module, three other modules include Expansion-Squeeze-Excitation (ESE) attention [64], Efficient Channel Attention (ECA) [65], and Convolutional Block Attention Module (CBAM) [66]. ESE examines how features interact when they are upsampled and downsampled and use an expansion step that may be used to extend modal and channel information. ECA utilizes a local cross-channel interaction technique that does not require dimensionality reduction and can be done quickly using 1D convolution. CBAM is a lightweight attention module that sequentially infers attention maps from a given feature map along two separate dimensions, channel and spatial, then the attention maps are multiplied to the input feature map for adaptive feature refinement. The performance of these modules was compared on the 16_channels dataset using the proposed architecture. Table 4 shows the results achieved. The best results are emboldened in all the following tables.

From the table above, SE attention module outperformed all other attention modules on all the evaluation metrics. Also, it is a lightweight attention module with minimal computational burden. Thus, we decided to include SE attention in the proposed architecture.

**Table 4** Segmentation results of using different attention layers in the proposed CropNet model on 16_channels dataset

| Attention module employed | Average accuracy (%) | Overall accuracy (%) | Macro-precision (%) | Macro-recall (%) | Macro-$F1$ score (%) | mIoU (%) |
|---|---|---|---|---|---|---|
| SE | 80.33 | 76.22 | 75.57 | 64.80 | 66.26 | 62.22 |
| ESE | 79.76 | 75.58 | 75.31 | 63.46 | 64.51 | 59.82 |
| ECA | 80.25 | 76.08 | 75.43 | 62.66 | 63.51 | 59.13 |
| CBAM | 78.73 | 74.93 | 75.25 | 61.70 | 63.01 | 58.95 |

### 3.2 Segmentation results on different datasets based on the number of channels

Table 5 presents the results of the segmentation task using the proposed model, CropNet, on all the four datasets: 7_channels, 9_channels, 13_channels, and 16_channels. These results were achieved with attention modules and feature fusion bars.

The 16_channels dataset, which included all the 13 Sentinel-2 bands as well as the computed indices, i.e. NDVI, NDWI, and Euclidean norm, outperformed the other three datasets: 7_channels, 9_channels, and 13_channels on all the evaluation metrics. The maximum OA achieved is 76.22% which is 3.26% higher as compared to the second-best results on the 9_channels and 13_channels dataset. Similarly, the mIoU, which is an important semantic segmentation metric, is also improved on the 16_channels dataset to 62.22% as compared to 58.96% on the 9_channels dataset.

The 13_channels dataset which contains Sentinel-2 bands only and no spectral indices performed on par with 7_channels dataset and 9_channels dataset but not with 16_channels dataset on various evaluation metrics but mIoU, the most important metric for segmentation. Still, it wasn't any more effective than them. The mIoU achieved is 55.64% and 5.68% lower than that achieved on 16_channels dataset. It indicates the importance of feeding spectral indices if the goal of a deep learning model is to analyse vegetation-related information, such as crop classification, land cover mapping, or vegetation health monitoring. The model can learn to extract relevant patterns and relationships from these indices along with other available

**Table 5** Segmentation results of CropNet on 7_channels, 9_channels, 13_channels, and 16_channels datasets achieved with attention modules and feature fusion bars

| Evaluation Metric (%) | Dataset | | | |
|---|---|---|---|---|
| | 7_channels | 9_channels | 13_channels | 16_channels |
| Average accuracy | 78.19 | 78.52 | 76.07 | 80.33 |
| Overall accuracy | 71.52 | 72.96 | 72.96 | 76.22 |
| Macro-precision | 71.43 | 71.65 | 72.90 | 75.57 |
| Macro-recall | 61.19 | 61.71 | 60.03 | 64.80 |
| Macro-$F1$ score | 60.70 | 62.27 | 60.68 | 66.26 |
| mIoU | 58.38 | 58.96 | 55.64 | 62.22 |

data, such as satellite images or environmental parameters. By providing additional features such as spectral indices as input, the deep learning model can potentially enhance its ability to understand vegetation dynamics and make more accurate predictions. Thus, it can be concluded that including a greater number of Sentinel-2 bands and spectral indices provides important discriminating information to the classifier as compared to using only a fewer number of them.

### 3.3 Ablation study: analysing the effect of using attention modules and the feature fusion bars on the model's performance

We conducted an ablation study to systematically analyse and evaluate the contribution of individual design choices on the model's performance to gain a deeper understanding of the effectiveness of each modification in the CropNet architecture and help identify the key factors driving its improved performance. Specifically, we analysed the effects of using the chosen SE attention modules and the feature fusion bars technique on the performance of CropNet. Table 6 presents the results of (i) CropNet without attention modules and feature fusion bars, (ii) CropNet after adding attention modules but without feature fusion bars, and (iii) CropNet with attention and feature fusion bars. These results were obtained on the 16_channel dataset.

From Table 6, it can be seen that the attention modules and the feature fusion technique proved beneficial for segmentation. Overall accuracy was improved by 1.43% by the addition of attention modules to the decoder nodes of CropNet. This helped the model to focus on the extraction of essential features. The application of the feature fusion bars further improved the segmentation accuracy by 1.75%. The model's performance on other evaluation metrics also improved with the application of the attention modules and the feature fusion bars technique.

### 3.4 Further analysis of the proposed CropNet model

For gaining further insights into the performance of the proposed model, CropNet, we present the segmentation results on each crop in the dataset in Table 7. The overall percentage of pixels in the dataset belonging to each class of crops is given in the table. The dataset exhibits a high class imbalance as seen by the presence of poorly

**Table 6** Analysing improvement in the performance of the CropNet model after employing attention modules and the feature fusion bars technique on 16_channel dataset

| Evaluation Metric (%) | Without attention, without feature fusion bars | With attention, without feature fusion bars | With attention, with feature fusion bars |
|---|---|---|---|
| Average accuracy | 78.91 | 79.34 | 80.33 |
| Overall accuracy | 73.04 | 74.47 | 76.22 |
| Macro-precision | 73.49 | 73.08 | 75.57 |
| Macro-recall | 62.40 | 63.03 | 64.80 |
| Macro-$F1$ score | 62.27 | 62.75 | 66.26 |
| mIoU | 59.66 | 60.41 | 62.22 |

**Table 7** Performance of CropNet model on segmentation of individual crops

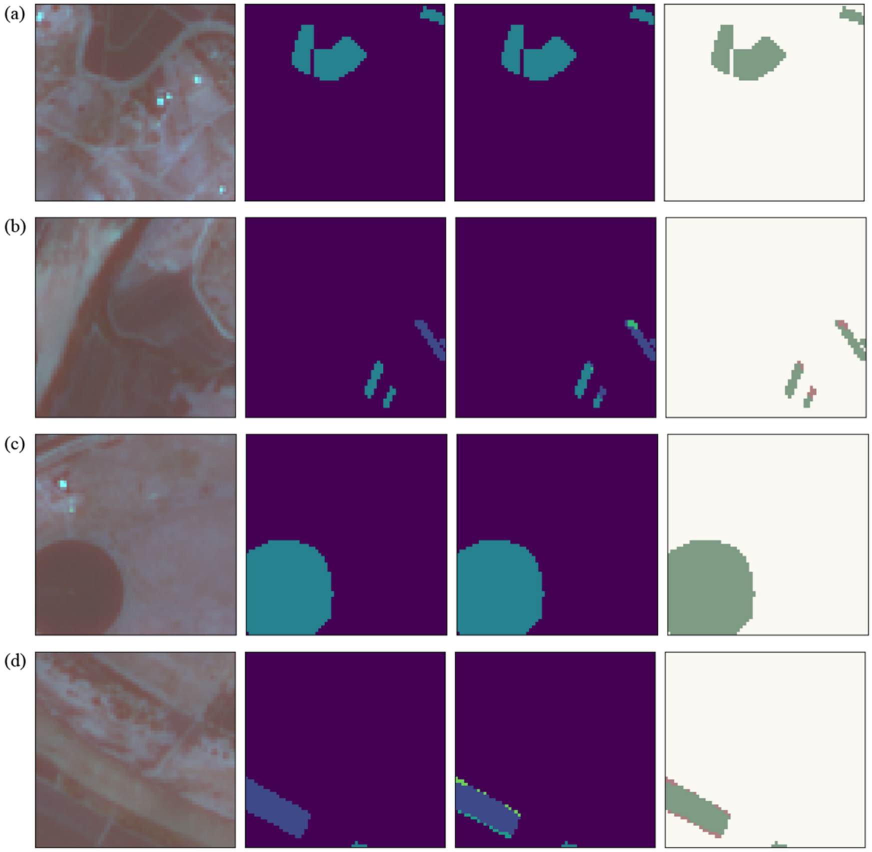
| Evaluation metrics (%) | Cotton | Vineyard | Grass | Lucern | Maize | Pecan | Vacant |
|---|---|---|---|---|---|---|---|
| Pixels belonging to the class (%) | 4.83 | 31.93 | 5.91 | 25.23 | 13.02 | 8.43 | 10.65 |
| Accuracy | 72.82 | 90.15 | 58.14 | 90.64 | 89.60 | 86.70 | 74.22 |
| Precision | 95.91 | 91.73 | 70.27 | 76.23 | 76.73 | 75.76 | 42.34 |
| Recall | 45.74 | 83.85 | 16.73 | 90.84 | 82.96 | 75.63 | 57.83 |
| $F1$ score | 61.94 | 87.61 | 27.02 | 82.89 | 79.72 | 75.70 | 48.89 |
| IoU | 44.87 | 77.96 | 15.62 | 70.79 | 66.28 | 60.90 | 32.35 |

represented crops such as ‘cotton’ and ‘grass’ with, respectively, 4.83% and 5.91% of the total pixels belonging to them. On the other hand, ‘vineyard’ and ‘lucern’ are the majority classes with, respectively, 31.93% and 25.23% of the total pixels belonging to these classes. The model demonstrated better performance on the majority classes with the highest accuracy of 90.64% shown by the ‘lucern’ class and the best IoU score of 77.96% yielded on the ‘vineyard’ class. These two classes demonstrated good performance on other evaluation metrics also, except for precision in which the ‘cotton’ class gave the best results of 95.91%, but its IoU score of 44.87% was low. The lowest performance was demonstrated by the ‘grass’ class with an accuracy of 58.14% and an IoU score as low as 15.62%. The performance of the model on the ‘vacant’ class was also weak with 32.35% IoU. The model demonstrated reasonable performance on other classes such as ‘maize’ and ‘pecan’ having accuracies of 89.60% and 86.70%, respectively.

Figure 7 displays some of the crop masks produced by the proposed model. From these qualitative results, it can be seen that the model is capable of segmenting different crops precisely producing accurate crop masks. It can also be seen that the model performs relatively poorly on field boundaries and fields with non-standard shapes, as shown in Fig. 7b, d.

### 3.5 Comparison with existing approaches

For validating the performance of our proposed methodology, we compared the results of our model with existing mainstream semantic segmentation models, U-Net [67] and U-Net++ [59], Deeplab v3+ [68], SegNet [69], and Attention U-Net [70] and also with those reported by [47], who also worked on the same ‘2019 Zindi’s Farm Pin Crop Detection’ dataset. These approaches have given excellent performance in other applications including segmentation of roads, biomedical images, and natural images. All these approaches form the basis of most of the CNN-based segmentation architectures used nowadays, thus making it a comprehensive comparison. These results are reported on pixel-wise classification using overall accuracy, macro-$F1$, and mIoU and on field-wise classification performance using overall accuracy and macro-$F1$. Table 8 shows the



Satellite image

Ground truth mask

Predicted mask

Similarities (in green) and differences (in red) between ground truth mask and predicted mask

**Fig. 7** Crop segmentation masks produced by CropNet

comparison of the results achieved on our proposed CropNet model with the existing approaches.

CropNet beats the other methods on all the evaluation metrics in pixel-wise classification as well as field-wise classification. On pixel-wise classification, CropNet enhances the overall accuracy, the macro-$F1$ score, and the mIoU by a margin of 3.18%, 5.62%, and 2.44% over the best results produced by other models. On field-wise classification, CropNet achieved an overall accuracy of 78.06%, which is 0.62% higher than that reported by [47] and 0.51% higher than those achieved with U-Net++. Similarly, our proposed model attained a macro-$F1$ score of 67.30% on field-wise classification, which is 1.3% greater than the earlier work by [47] and 4.54% higher than that attained with U-Net++.

**Table 8** Performance comparison of the proposed model, CropNet with the existing work

| | Pixel-wise classification results | | | Field-wise classification results | |
|---|---|---|---|---|---|
| | Overall accuracy | Macro-$F1$ | mIoU | Overall accuracy | Macro-$F1$ |
| U-Net [67] | 72.02 | 59.03 | 58.14 | 70.83 | 58.96 |
| U-Net++ [59] | 73.04 | 59.66 | 59.66 | 77.5 | 62.76 |
| Matvienko et al., 2022 [47] | 70.1 | 57 | – | 77.44 | 66 |
| Deeplab v3+ [68] | 71.81 | 60.64 | 59.78 | 77.39 | 59.20 |
| SegNet [69] | 71.35 | 56.56 | 54.36 | 72.87 | 55.54 |
| Attention U-Net [70] | 72.58 | 59.27 | 58.84 | 73.66 | 61.02 |
| CropNet (ours) | 76.22 | 66.26 | 62.22 | 78.06 | 67.30 |

## 4 Conclusions

The experimental results demonstrate the effectiveness of our proposed CropNet model on crop type detection and segmentation in the difficult scenario of smallholder farms. This can be attributed to the U-Net++ based structure of CropNet which helps to enhance the semantic similarity of the feature maps between the encoder and the decoder. Furthermore, the addition of the attention modules in the decoder nodes helps to pay attention to the most relevant features of a feature map. The feature fusion technique facilitates getting a more precise prediction by bringing the spatial location information from the previous layers. All these features helped the model to gain superior results even with a limited size of the training samples. The proposed model achieved an overall accuracy and a macro-$F1$ of 78.06% and 67.30%, respectively, on field-wise classification, thus improving the state of the art by 0.51% and 1.3%, respectively. On pixel-wise semantic segmentation, our model improved the mIoU by 2.56%. The performance results thus demonstrate that our model is computationally efficient for the task of crop type detection from satellite images of small field parcels.

The trained model can be used for the extraction of fields belonging to different crops from the satellite images. Thus, it can help by providing a basis for crop yield forecasting and ultimately contributing to achieving food security, one of the 17 SDGs of the United Nations' 2030 Agenda for Sustainable Development. It can help the government, farmers, buyers, and other stakeholders for making necessary strategies and arrangements for production, harvesting, procurement, stocking, marketing, etc. It can also help give information about crop diversity, crop growth patterns, and trends and facilitate crop monitoring.

Although the proposed architecture is trained on a small dataset, the dataset is diverse and a representative of a major agricultural region in South Africa. This region has been drought-stricken during the recent years. In 2021, 20.2% of Africans were severely food insecure. For countries aiming at boosting food security and agricultural growth, improved models for crop type automatic forecast and activity monitoring are essential. Our proposed model is such an effort in this direction.

Generalization is an important factor for the practical application and adoption of any deep learning architecture. Due to the location-specific nature of crop phenotypic and phenology information as well as the variations in canopy-level spectral reflectance among various environments and management practises, crop type detection models calibrated for one region cannot be easily generalized to another region. However, the proposed trained model may be tested for its transferability on other study sites with different geographical characteristics to compare and analyse its performance in the future research.

In the proposed work, we stacked all the timepoints together in the images' channels which does not allow the model to properly learn from the patterns in the imagery through time. In the future, recurrent networks such as temporal convolutional networks can be explored to learn temporal patterns. A more sophisticated loss function such as boundary loss may be used to minimize the error at field boundaries. Also, a model for estimating yield prediction from the detected crop fields can be devised.

**Authors contributions** PKB performed the experiments and wrote the main manuscript text. MKS supervised the research work and reviewed the manuscript.

**Funding** Partial financial support was received from ESA (European Space Agency) Network of Resources Initiative.

**Data availability** The datasets analysed during the current study are available at https://zindi.africa/competitions/farm-pin-crop-detection-challenge/data.

## Declarations

**Competing interests** The authors have no competing interests to declare that are relevant to the content of this article.

**Ethical approval** This declaration is not applicable.

## References

1. You J, Li X, Low M, et al (2017) Deep gaussian process for crop yield prediction based on remote sensing data. In: Proceedings of the 31st AAAI Conference on Artificial Intelligence (AAAI-17). pp 4559–4565
2. Rembold F, Atzberger C, Savin I, Rojas O (2013) Using low resolution satellite imagery for yield prediction and yield anomaly detection. Remote Sens Environ 5:1704–1733. https://doi.org/10.3390/rs5041704
3. Mohanty SP, Czakon J, Kaczmarek KA et al (2020) Deep learning for understanding satellite imagery: an experimental survey. Front Artif Intell 3:1–21. https://doi.org/10.3389/frai.2020.534696
4. Wang P, Chen P, Yuan Y, et al (2018) Understanding convolution for semantic segmentation. In: 2018 IEEE Winter Conference on Applications of Computer Vision (WACV). pp 1451–1460
5. Quarmby NA, Milnes M, Hindle TL, Silleos N (1993) The use of multi-temporal NDVI measurements from AVHRR data for crop yield estimation and prediction. Int J Remote Sens 14:199–210. https://doi.org/10.1080/01431169308904332
6. Johnson DM (2014) An assessment of pre- and within-season remotely sensed variables for forecasting corn and soybean yields in the United States. Remote Sens Environ 141:116–128. https://doi.org/10.1016/j.rse.2013.10.027

7. Bolton DK, Friedl MA (2013) Forecasting crop yield using remotely sensed vegetation indices and crop phenology metrics. Agric For Meteorol 173:74–84. https://doi.org/10.1016/j.agrformet.2013.01.007
8. Kuwata K, Shibasaki R (2015) Estimating crop yields with deep learning and remotely sensed data. In: 2015 IEEE International Geoscience and Remote Sensing Symposium (IGARSS). IEEE, pp 858–861
9. Satir O, Berberoglu S (2016) Crop yield prediction under soil salinity using satellite derived vegetation indices. Field Crops Res 192:134–143. https://doi.org/10.1016/j.fcr.2016.04.028
10. Kim N, Lee Y-W (2016) Machine learning approaches to corn yield estimation using satellite images and climate data: a case of Iowa State. J Korean Soc Surv Geod Photogramm Cartogr 34:383–390
11. Asgarian A, Soffianian A, Pourmanafi S (2016) Crop type mapping in a highly fragmented and heterogeneous agricultural landscape. Comput Electron Agric 127:531–540. https://doi.org/10.1016/j.compag.2016.07.019
12. Gilbertson JK, Kemp J, van Niekerk A (2017) Effect of pan-sharpening multi-temporal landsat 8 imagery for crop type differentiation using different classification techniques. Comput Electron Agric 134:151–159. https://doi.org/10.1016/j.compag.2016.12.006
13. Kang J, Zhang H, Yang H, Zhang L (2018) Support vector machine classification of crop lands using sentinel-2 imagery. In: 2018 7th International Conference on Agro-geoinformatics (Agro-geoinformatics). pp 1–6
14. Kumar P, Prasad R, Choudhary A et al (2017) A statistical significance of differences in classification accuracy of crop types using different classification algorithms. Geocarto Int 32:206–224. https://doi.org/10.1080/10106049.2015.1132483
15. Lussem U, Hütt C, Waldhoff G (2016) Combined analysis of sentinel-1 and rapideye data for improved crop type classification: an early season approach for rapeseed and cereals. ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences XLI-B8:959–963. https://doi.org/10.5194/isprs-archives-XLI-B8-959-2016
16. Zheng B, Myint SW, Thenkabail PS, Aggarwal RM (2015) A support vector machine to identify irrigated crop types using time-series Landsat NDVI data. Int J Appl Earth Obs Geoinf 34:103–112. https://doi.org/10.1016/j.jag.2014.07.002
17. Khatami R, Mountrakis G, Stehman SV (2016) 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 177:89–100. https://doi.org/10.1016/j.rse.2016.02.028
18. Shukla G, Garg RD, Srivastava HS, Garg PK (2018) Performance analysis of different predictive models for crop classification across an aridic to ustic area of Indian states. Geocarto Int 33:240–259. https://doi.org/10.1080/10106049.2016.1240721
19. Chen Y, Lu D, Moran E et al (2018) Mapping croplands, cropping patterns, and crop types using MODIS time-series data. Int J Appl Earth Obs Geoinf 69:133–147. https://doi.org/10.1016/j.jag.2018.03.005
20. Schultz B, Immitzer M, Formaggio AR et al (2015) Self-guided segmentation and classification of multi-temporal landsat 8 images for crop type mapping in Southeastern Brazil. Remote Sens (Basel) 7:14482–14508. https://doi.org/10.3390/rs71114482
21. Vuolo F, Neuwirth M, Immitzer M et al (2018) How much does multi-temporal Sentinel-2 data improve crop type classification? Int J Appl Earth Obs Geoinf 72:122–130. https://doi.org/10.1016/j.jag.2018.06.007
22. Rußwurm M, Pelletier C, Zollner M, et al (2020) Breizhcrops: A time series dataset for crop type mapping. In: International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives. pp 1545–1551
23. Russakovsky O, Deng J, Su H et al (2015) ImageNet Large Scale Visual Recognition Challenge. Int J Comput Vis 115:211–252. https://doi.org/10.1007/s11263-015-0816-y
24. Kussul N, Lavreniuk M, Skakun S, Shelestov A (2017) Deep learning classification of land cover and crop types using remote sensing data. IEEE Geosci Remote Sens Lett 14:778–782. https://doi.org/10.1109/LGRS.2017.2681128
25. Zhong L, Hu L, Zhou H (2019) Deep learning based multi-temporal crop classification. Remote Sens Environ 221:430–443. https://doi.org/10.1016/j.rse.2018.11.032

26. Cai Y, Guan K, Peng J et al (2018) A high-performance and in-season classification system of field-level crop types using time-series Landsat data and a machine learning approach. Remote Sens Environ 210:35–47. https://doi.org/10.1016/j.rse.2018.02.045
27. Ji S, Zhang C, Xu A et al (2018) 3D convolutional neural networks for crop classification with multi-temporal remote sensing images. Remote Sens (Basel) 10:1–17. https://doi.org/10.3390/rs10010075
28. Terliksiz AS, Altýlar DT (2019) Use of deep neural networks for crop yield prediction: a case study of soybean yield in Lauderdale County, Alabama, USA. In: 2019 8th International Conference on Agro-Geoinformatics (Agro-Geoinformatics). pp 1–4
29. Wang S, Di Tommaso S, Faulkner J et al (2020) Mapping crop types in southeast india with smartphone crowdsourcing and deep learning. Remote Sens (Basel) 12:1–42. https://doi.org/10.3390/rs12182957
30. Rußwurm M, Körner M (2017) Multi-temporal land cover classification with long short-term memory neural networks. Int Arch Photogram Remote Sens Spat Inf Sci XLII-1/W1:551–558. https://doi.org/10.5194/isprs-archives-XLII-1-W1-551-2017
31. Ndikumana E, Ho Tong Minh D, Baghdadi N et al (2018) Deep recurrent neural network for agricultural classification using multitemporal SAR Sentinel-1 for Camargue, France. Remote Sens (Basel) 10:1–16. https://doi.org/10.3390/rs10081217
32. Waldner F, Diakogiannis FI (2020) Deep learning on edge: Extracting field boundaries from satellite images with a convolutional neural network. Remote Sens Environ 245:111741. https://doi.org/10.1016/j.rse.2020.111741
33. Rustowicz R, Cheong R, Wang L, et al (2019) Semantic segmentation of crop type in Africa: a novel dataset and analysis of deep learning methods. In: CVPR Workshops. pp 75–82
34. Tai Y, Yang J, Liu X (2017) Image super-resolution via deep recursive residual network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp 3147–3155
35. Jiao L, Dong S, Zhang S et al (2020) AF-RCNN: An anchor-free convolutional neural network for multi-categories agricultural pest detection. Comput Electron Agric 174:1–9. https://doi.org/10.1016/j.compag.2020.105522
36. Dong S, Wang R, Liu K et al (2021) CRA-Net: a channel recalibration feature pyramid network for detecting small pests. Comput Electron Agric 191:1–8. https://doi.org/10.1016/j.compag.2021.106518
37. Jiao L, Xie C, Chen P et al (2022) Adaptive feature fusion pyramid network for multi-classes agricultural pest detection. Comput Electron Agric 195:1–9. https://doi.org/10.1016/j.compag.2022.106827
38. Li R, Wang R, Xie C et al (2022) A multi-branch convolutional neural network with density map for aphid counting. Biosyst Eng 213:148–161. https://doi.org/10.1016/j.biosystemseng.2021.11.020
39. Dong S, Du J, Jiao L et al (2022) Automatic crop pest detection oriented multiscale feature fusion approach. Insects 13:1–17. https://doi.org/10.3390/insects13060554
40. Samberg LH, Gerber JS, Ramankutty N et al (2016) Subnational distribution of average farm size and smallholder contributions to global food production. Environ Res Lett 11:124010. https://doi.org/10.1088/1748-9326/11/12/124010
41. Khan HR, Gillani Z, Jamal MH et al (2023) Early identification of crop type for smallholder farming systems using deep learning on time-series sentinel-2 imagery. Sensors 23:1. https://doi.org/10.3390/s23041779
42. Lowder SK, Skoet J, Raney T (2016) The number, size, and distribution of farms, smallholder farms, and family farms worldwide. World Dev 87:16–29. https://doi.org/10.1016/j.worlddev.2015.10.041
43. Ricciardi V, Ramankutty N, Mehrabi Z et al (2018) How much of the world's food do smallholders produce? Glob Food Sec 17:64–72. https://doi.org/10.1016/j.gfs.2018.05.002
44. Potgieter AB, Zhao Y, Zarco-Tejada PJ et al (2021) Evolution and application of digital technologies to predict crop type and crop phenology in agriculture. In Silico Plants 3:1–23. https://doi.org/10.1093/insilicoplants/diab017
45. Yu L, Wang J, Clinton N et al (2013) FROM-GC: 30 m global cropland extent derived through multisource data integration. Int J Digit Earth 6:521–533. https://doi.org/10.1080/17538947.2013.822574
46. Xiong J, Thenkabail PS, Tilton JC et al (2017) Nominal 30-m cropland extent map of continental Africa by integrating pixel-based and object-based algorithms using sentinel-2 and landsat-8 data on google earth engine. Remote Sens (Basel) 9:1–27. https://doi.org/10.3390/rs9101065

47. Matvienko I, Gasanov M, Petrovskaia A et al (2022) Bayesian aggregation improves traditional single image crop classification approaches. Sensors 22:1–13. https://doi.org/10.3390/s22228600
48. Liu S, Li M, Zhang Z et al (2020) Ground-based cloud classification using task-based graph convolutional network. Geophys Res Lett 47:1–8. https://doi.org/10.1029/2020GL087338
49. Yu L, Wang J, Li X et al (2014) A multi-resolution global land cover dataset through multisource data aggregation. Sci China Earth Sci 57:2317–2329. https://doi.org/10.1007/s11430-014-4919-z
50. Long J, Shelhamer E, Darrell T (2015) Fully convolutional networks for semantic segmentation. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp 3431–3440
51. Vaswani A, Shazeer N, Parmar N, et al (2017) Attention Is All You Need. In: 31st Conference on Neural Information Processing Systems (NIPS 2017). pp 1–11
52. Dai Y, Yu J, Zhang D et al (2022) RODFormer: high-precision design for rotating object detection with transformers. Sensors 22:1–13. https://doi.org/10.3390/s22072633
53. Deng Z, Zhou B, He P, et al (2022) A position-aware transformer for image captioning. Comput Mater Continua 70:2065–2081. https://doi.org/10.32604/cmc.2022.019328
54. Xu Z, Zhang W, Zhang T et al (2021) Efficient transformer for remote sensing image segmentation. Remote Sens (Basel) 13:1–24. https://doi.org/10.3390/rs13183585
55. Zhang C, Jiang W, Zhang Y et al (2022) Transformer and CNN hybrid deep neural network for semantic segmentation of very-high-resolution remote sensing imagery. IEEE Trans Geosci Remote Sens 60:1–20. https://doi.org/10.1109/TGRS.2022.3144894
56. Ulku I, Akagündüz E (2022) A survey on deep learning-based architectures for semantic segmentation on 2D images. Appl Artif Intell 00:1–45. https://doi.org/10.1080/08839514.2022.2032924
57. Palchoudhuri Y, Valcarce-Diñeiro R, King P, Sanabria-Soto M (2018) Classification of multi-temporal spectral indices for crop type mapping: a case study in Coalville, UK. J Agric Sci 156:1–13. https://doi.org/10.1017/S0021859617000879
58. Gao B (1996) NDWI—a normalized difference water index for remote sensing of vegetation liquid water from space. Remote Sens Environ 58:257–266. https://doi.org/10.1016/S0034-4257(96)00067-3
59. Zhou Z, Rahman Siddiquee MM, Tajbakhsh N, Liang J (2018) UNet++: A nested U-net architecture for medical image segmentation. In: Stoyanov D, Taylor Z, Carneiro G et al (eds) Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support. Springer International Publishing, Cham, pp 3–11
60. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. pp 1–9
61. Hu J, Shen L, Albanie S et al (2020) Squeeze-and-Excitation Networks. IEEE Trans Pattern Anal Mach Intell 42:2011–2023. https://doi.org/10.1109/TPAMI.2019.2913372
62. Lin T-Y, Goyal P, Girshick R, et al (2017) Focal Loss for Dense Object Detection. In: 2017 IEEE International Conference on Computer Vision (ICCV). pp 2999–3007
63. Gonzales C, Sakla W (2019) Semantic Segmentation of Clouds in Satellite Imagery Using Deep Pre-trained U-Nets. In: Proceedings - Applied Imagery Pattern Recognition Workshop. pp 1–7
64. Shu X, Yang J, Yan R, Song Y (2022) Expansion-squeeze-excitation fusion network for elderly activity recognition. IEEE Trans Cir and Sys for Video Technol 32:5281–5292. https://doi.org/10.1109/TCSVT.2022.3142771
65. Wang Q, Wu B, Zhu P, et al (2020) ECA-Net: Efficient channel attention for deep convolutional neural networks. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. pp 11531–11539
66. Woo S, Park J, Lee JY, Kweon IS (2018) CBAM: Convolutional block attention module. In: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). pp 3–19
67. Ronneberger O, Fischer P, Brox T (2015) U-Net: Convolutional Networks for Biomedical Image Segmentation. In: Navab N, Hornegger J, Wells WM, Frangi AF (eds) Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015. Springer International Publishing, Cham, pp 234–241
68. Chen Liang-Chieh and Zhu Y and PG and SF and AH (2018) Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation. In: Ferrari Vittorio and Hebert M and SC and WY (ed) Computer Vision – ECCV 2018. Springer International Publishing, Cham, pp 833–851
69. Badrinarayanan V, Kendall A, Cipolla R (2017) SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. IEEE Trans Pattern Anal Mach Intell 39:2481–2495. https://doi.org/10.1109/TPAMI.2016.2644615

70. Oktay O, Schlemper J, Folgoc L Le, et al (2018) Attention U-Net: Learning Where to Look for the Pancreas. In: 1st Conference on Medical Imaging with Deep Learning (MIDL 2018). Amsterdam, The Netherlands

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.