Deep learning models to map an agricultural expansion area with MODIS and Sentinel-2 time series images

Dong Luo,[a,*] Marcellus M. Caldas,[a] Huichen Yang,[b]
[a]Kansas State University, Department of Geography and Geospatial Sciences, Manhattan, KS, United States, 66506
[b]Kansas State University, Department of Computer Science, Manhattan, KS, United States, 66506

Abstract. Mapping changing land use and land cover is important for land management and environment analysis. In this study, we tried to build deep learning models to classify land use and land cover over time at an agricultural expansion area in Matopiba region, Brazil with MCD43A4 V006 MODIS and Sentinel-2 Multispectral Instrument (MSI) time series data. We collected time series MODIS data and Sentinel-2 A/B MSI data from 2015 to 2020, and prepared overlaying small patches with containing blue, green, red, NIR, and SWIR-1 bands as features. Then the both datasets were used to build and train the CNN model, the CNN-GRU model and the CNN-LSTM model, respectively. We evaluated these three trained models with ground truth data, and the CNN-LSTM model (overall accuracy: 91.29% from MODIS data and 89.47% from Sentinel-2 data) was better than the CNN-GRU model (overall accuracy: 89.19% from MODIS data and 88.61% from Sentinel-2 data) and the CNN model (overall accuracy: 89.17% from MODIS data and 86.02% from Sentinel-2 data). Our results also showed that the accuracy from cropland and savanna classes were higher than grassland and forest classes in all three models. These two classes generated from the CNN-LSTM model performed better than the other two deep learning models. The results from these two datasets indicated that the methods were reliable for both coarse and medium spatial resolution satellite images and time series remote sensing images worked better than single image for classification problems when considering land use and land cover change over time. The results also provided an alternative way to prepare input data from satellite images for deep learning models. Furthermore, the classification results of the whole agricultural expansion area captured major land use and land cover and it can be used as additional dataset for further environmental analysis at a regional scale.

Keywords: land use and land cover, deep learning, remote sensing, MODIS, Sentinel-2, agriculture.

1 Study area and data

1.1 Study area

The study area was located in the Matopiba agricultural frontier of Brazil. The Matopiba is a geographic Savanna (Cerrado) region across the states of Maranhão, Tocantins, Piaui, and Bahia; an area equivalent to twice the size of Germany and almost three times the size of the United Kingdom, encompassing 324 thousand rural properties and 6 million inhabitants [48]. This new frontier is identified as one of the few Cerrado areas with available untapped land suitable for agricultural production in Brazil. It has two seasons. From April to September is the dry season, and the wet season is from October to March. During the last three decades, this region has been experienced enormous agricultural expansion with more than 50% of its natural vegetation converted into agriculture areas [49,50]. In this study, we were interested in the east side of the Matopiba region (around 425,666 km^2) which is a traditional agricultural expansion area and a typical savanna environment (Fig.1). Interestingly, the study area was also covered by total 60 Sentinel-2 tiles, and we created training and test data based on these tiles. Specifically, we chose three tiles area (23LLF, 23LMH, 23LNF) as training sites, and tile 23 LNK area as test site for MODIS purpose. These tiles were chosen based on the rule that each site should cover all types of target classes. Each site is around 11,567 km^2 with 225 by 225 pixels of MODIS data (Fig. 1). In addition, within these chosen tiles, we selected two tiles (23LMH and 23LNF) as training data for the Sentinel-2 purpose. Notably, we just chose the quarter (left top part) of the entire tile (23LNK) as our test data for the Sentinel-2 purpose, because it is the only place having cropland and the goal of this study focused on mapping agricultural expansion area.

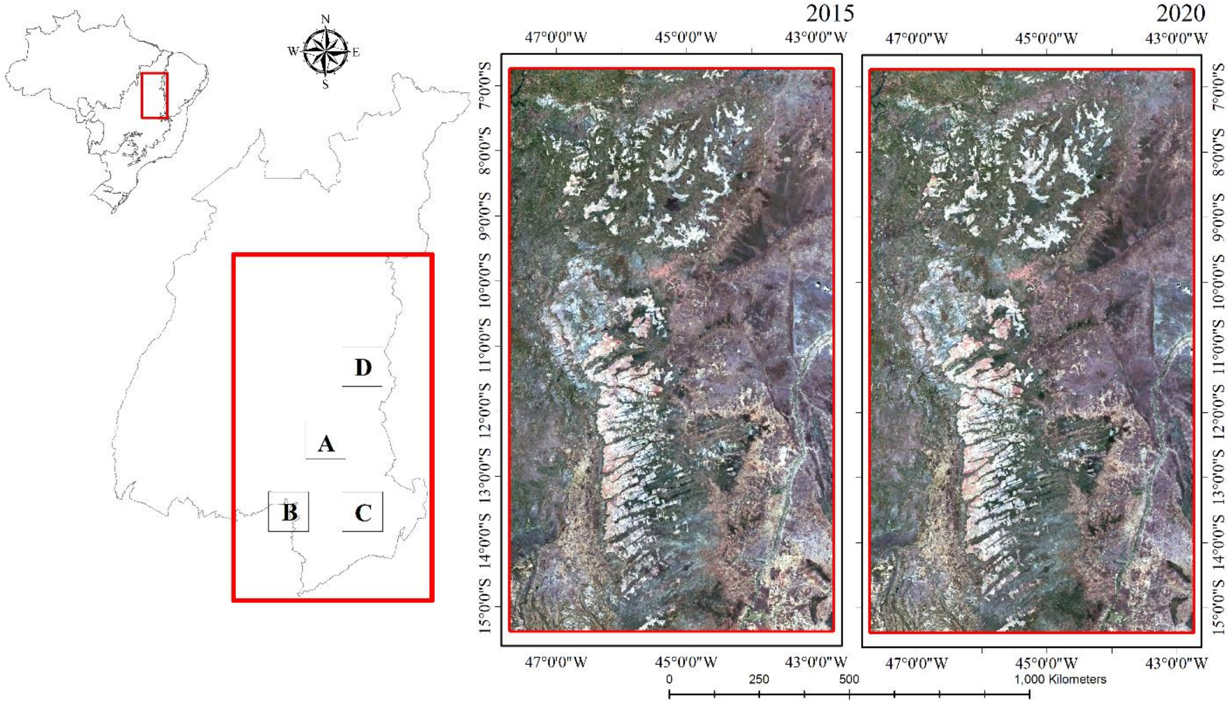

Fig. 1 The geographic location of the study area and three training sites (A, B and C) with one testing site D. MODIS RGB images at the right side (including coordinate information and stretch type is standard deviations) illustrate the land use and land cover change in 2015 and 2020 in September.

1.2 Data source

To fully cover the interested areas, we first used daily MCD43A4 V006 MODIS product with spatial resolution of 478m by 478m. We set study time from 2015 to 2020, due to the newest available LULC map in this area is 2020. The product used Nadir Bidirectional Reflectance Distribution Function (BRDF) – adjusted reflectance, which already removed the view angle effects caused by the sensor to avoid some potential biases [51]. The study area was covered by the MCD43A4 V006 product with h13v10 and h13v09 tiles. Since we wanted to track its annual changes, we chose September (Julian day from 244-274) as our data source (with total 360 HDF format images) (Table 1) to avoid potential cloud problem in the wet season that could affect the quality of the data processing. MCD43A4 V006 product has 7 shortwave bands (each band also has one quality band), with band 1 (620-670 mm), band 2 (841-876 mm), band 3 (459-479 mm), band 4 (545-565 mm), band 5 (1230-1250 mm), band 6 (1628-1652 mm), and band 7 (2105-2155 mm). We selected band 1, band 2, band 3, band 4, and band 5 that contained blue, green, red, near infrared (NIR) band and shortwave infrared (SWIR-1) band as our features (Table 2) because these bands are good for vegetation research [52]. We used each quality band to separate good quality pixels and set bad quality pixels as 0, then all 5 bands stacked as one image. Since we focused on annual changes, we generated one image per each year from all 30 images in September by calculating mean value of each pixel.

Table 1 Total number of images used in the study

Year	MCD43A4 V006	Sentinel-2 23LMH	Sentinel-2 23LNF	Sentinel-2 23LNK
Sum	360	13	15	12
September, 2015	60	0	0	0
September, 2016	60	1	1	1
September, 2017	60	1	3	3
September, 2018	60	4	5	1
September, 2019	60	3	3	2
September, 2020	60	4	4	5

We also prepared Sentinel-2 Multispectral Instrument (MSI) time series data to test the flexibility of deep learning models. Compared with the MODIS products, Sentinel-2 data has much finer spatial resolution (10m, 20m, and 60m). The satellite is the constellation of Sentinel-2A and 2B that can produce 2~5 days revisit interval at the same spot globally. Currently, the users can access L1C (top of atmosphere product) and L2A (bottom of atmosphere product) data. We downloaded all available chosen Sentinel-2 tiles (L1C product) with cloud cover less than 20% and used Sen2Cor v2.10 software [53] to generate bottom of atmosphere (L2A) data from 2018 to 2020. For data from 2016 and 2017, we used Sen2Cor v2.09 software to generate bottom of atmosphere data due to European Space Agency (ESA) used old Product Specification Document (PSD) that cannot process by Sen2Cor v2.10 software. We chose L2A data because it has been atmospherically corrected for each pixel. The same as the MODIS time series data, we just acquired all available images within September from 2015 to 2020. Notably, some Sentinel-2 images could only partially cover the entire tile area due to the edge or top/bottom of the Data strip, and we just kept those tiles that useable data was higher than 65%. The total number of images used in the study was shown in Table 1. Although the data has variety of bands, to be in the line with the MODIS data, we selected band 2, band 3, band 4, band 8A, and band 11 in this work (Table 2). As the same as the MCD43A4 V006, we calculated each pixel's mean value with all available images (more than 1 image) per each tile and per each year. Importantly, since the Sentinel-2 L2A product contains Scene Classification Layer (SCL) that can help us filter out contaminated pixels (such as cloud, snow, etc.), we created clear pixels mask by summarizing all clear pixels from each SCL layer of the same tile from 2015 to 2020. Only these pixels that were clear in all SCL images were kept. Then, the mask served as a universal mask for the following steps.

Table 2 Datasets used in the study and its band distribution

Band name	MCD43A4 V006	Sentinel-2
Blue	Band 3 (459-479 mm)	Band 2 (459.8-524.0 mm)
Green	Band 4 (545-565 mm)	Band 3 (542.8-577.6 mm)
Red	Band 1 (620-670 mm)	Band 4 (649.3-679.9 mm)
NIR	Band 2 (841-876 mm)	Band 8A (854.5-875.0 mm)
SWIR-1	Band 5 (1230-1250 mm)	Band 11 (1568.7-1658.3 mm)

As a typical agricultural expansion area, cropland area in the study area was frequently changed from grassland, savanna, or forest these years [54]. Since we had MCD43A4 V006 and Sentinel-2 data, to create ground truth data for the deep learning models, we chose MODIS product MCD12Q1 V6 for MODIS purpose and Brazilian Annual Land Use and Land Cover Mapping Project (MapBiomas) Collection 6 (https://mapbiomas.org/en) for Sentinel-2 purpose. MODIS product MCD12Q1 V6 has global annual land use and land cover map (http://LPDAAC.usgs.gov) and we used International Geosphere-Biosphere Programme (IGBP) classification in this study. We chose these two products because they have been used in the remote sensing community [55,56]. To fit the goal of the study, we just kept pixels that didn't change their class type from 2015 to 2020 with MCD12Q1-IGBP and MapBiomas data, respectively. Importantly, each product could have different definition about each land use and land cover type [57], and we carefully compared each class and referred the existed literature of LULC schemes [58], to create target classes for this study. Finally, we have four classes with cropland, grassland, savanna, and forest. The explanation of each class can be found in the Table 3. In addition, to improve the quality of the ground truth data and reduce the bias from different datasets, we spatially interpolated MapBiomas data (30m spatial resolution) with nearest method to match pixel spatial resolution of the MCD12Q1 IGBP data. Then, we compared these two ground truth data and just kept those pixels having the same value in both data as the ground truth data for MODIS purpose. Finally, since MapBiomas data is a 30m spatial resolution product, we used the same spatial interpolated method to resample pixel size to 20 meter that is the same as the Sentinel-2 input data used in this research.

Table 3 Land use and land cover types used in the study and its description

Class name	Description	MCD12Q1-IGBP	MapBiomas-C6
Cropland	The area used for the production of annual crops	Croplands	Soybean, sugar cane, other temporary crops, coffee, other perennial crop, mosaic agriculture and pasture
Grassland	The areas dominated by grass types of vegetation and pasture	Grasslands	Grassland, pasture
Savanna	Grass covered area but interrupted by trees and trees cannot generate closed canopy [59]	Woody savannas, savannas	Savanna formation
Forest	Those areas that trees can form closed canopy	Evergreen needleleaf forests, Evergreen broadleaf forests, deciduous needleleaf forests, deciduous	Forest formation

	broadleaf forests, mixed forests

2 Methods

In this study, we used a CNN-LSTM model, a CNN-GRU model and a CNN model to classify LULC over time. The idea came from the CNN algorithm extracted features considering its spatial correlation with surrounding pixels, which can optimize features and reduce computing time, and the RNN neural networks (LSTM or GRU) were used to learn the changes of the same pixel over time. All models followed the same process: data preparation, classifier model achievement and evaluation, and classification.

2.1. Input data preparation

With the study area, we prepared multiple bands of remote sensing image data. We sequenced the total of 6 years MODIS data in the following order: 2015, 2016, 2017, 2018, 2019, and 2020, and Sentinel-2 data was from 2016 to 2020 due to there was zero image in 2015 (Table 1). Deep learning models need large volume of data to allow the model to fully learn the information. In this study, we used a patch-based method to create overlapping small patches from the remote sensing images [22,45]. Considering the spatial pattern of different classes, each small patch (15*15* 5) was created to represent height, width, and depth (or channel). One common drawback of this method was that patches at the edge of remote sensing image could cause potential bias by either using 0 to generate designed patch size or removing edge pixels [30,60]. We removed edge pixels to avoid potential bias during the model process. Notably, we also used the same method to process ground truth data and calculated the center pixel value as the LULC class for each small patch in this work.

2.2. The CNN model

The CNN is to use convolutional and pooling layers to extract the essential features from input image and uses those features to understand and classify the image. Specifically, the convolutional layer serves as a “filter” to slide the entire image for features extraction by downsizing the input image and increasing image dimension. The pooling layers then further reduce the size of the image to focus on the most important features thus training the network in a faster manner. Following the pooling layer is the fully connected layer that flattens the output of the polling layer into one-dimensional vector and generates a list of different possible labels corresponding to the image. In our approach, the CNN model contains four 2D convolutional layers, and two additional 2D max pooling layers added to the last two convolutional layers (Fig. 2). To avoid data missing, we used the padding method to get the same extent with the input data in the first and third convolutional layers. Considering the input (each patch) to each convolutional layer is 15 by 15 pixels, we set filters as 3*3 to take into account of each class pattern. Then the output from the second max pooling layer was flatten and connected with two dense layers with the second one predicted the results. In addition, to avoid overfitting, we applied dropout layer (set value was 0.5) after the first dense layer (Fig. 2). As shown in the Figure 2, after flattened, the parameter from the first dense was slightly different with MODSI data and Sentinel-2 data.

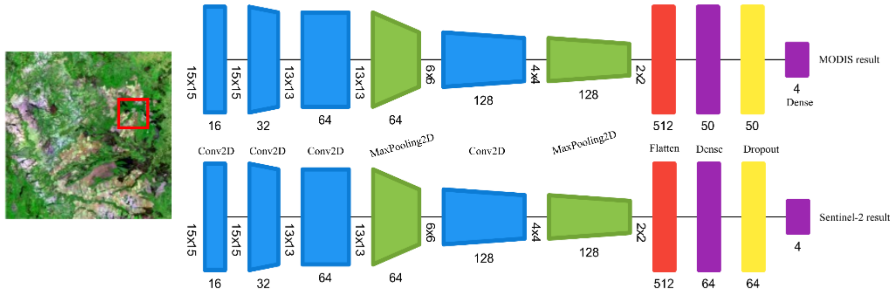

Fig. 2 The CNN models used in the study, top is the CNN model architecture for MODIS data, and bottom is the model for Sentinel-2 data. Red box is one small patch example.

2.3. The CNN-LSTM model

The created CNN architecture also served as a feature extraction part for the CNN-LSTM model (Fig. 3). During the feature extraction part, we used TimeDistributed function in Keras (https://keras.io/api/layers/recurrent_layers/time_distributed/) to wrap up time series small patches. Then we flattened the output as the input for the LSTM part to learn the changes of the same pixel over time. The LSTM has a long memory part, a short memory part and three different gates. These gates have two major functions: (1) to regulate the quantity of information and forget/remember during the process; (2) to deal with the problem of gradient disappearance/bursting problem [34]. There are different types of input and output relationship of the LSTM such as one-to-one, many-to-many and many-to-one. We used one LSTM with many-to-one type in this study. Finally, we flattened the output from the LSTM and followed two dense layers to generate the result (Fig. 3). As the same as the CNN model, we also applied a dropout layer (set value was 0.5) after the first dense layer to avoid overfitting issue of the model. A SoftMax layer was followed on the dense layer to predict the final multi-class result (Fig. 3). The SoftMax priority was given instead of the Sigmoid function, because the value of the SoftMax layer can be considered as a probability distribution on classes that total up to 1 [61].

The MODIS and Sentinel-2 data were shared the same CNN-LSTM model structure in this work (Fig. 3).

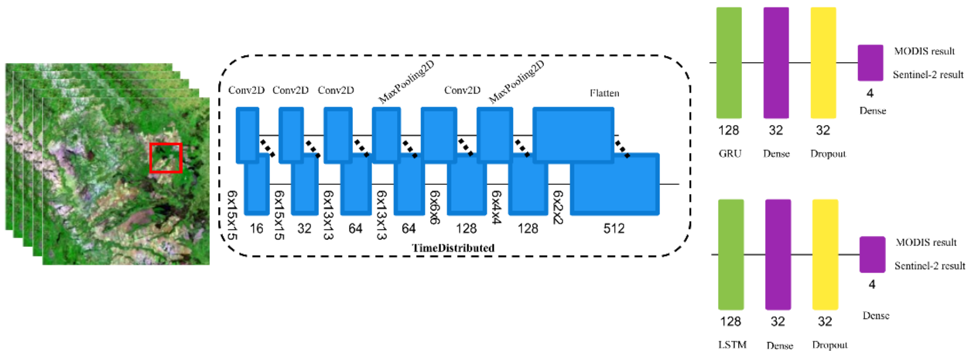

Fig. 3 The CNN-GRU model and the CNN-LSTM model used in the study, top is the CNN-GRU model architecture for both datasets, and bottom is the CNN-LSTM model for both datasets. Red box is one small patch example.

2.4. The CNN-GRU model

We also applied a CNN-GRU model which has similar architecture as the CNN-SLTM model for performance evaluation. Compared with the LSTM algorithm, GRU algorithm just has two gates, and they are reset gate and update gate [62]. Specifically, the update gate has the similar function to the forget and input gates of the LSTM and the reset gate was used to decide how much past information needs to be forgotten. We used the same CNN part as the CNN-LSTM model (Fig. 3). Then, we added one GRU layer and set output layer as just one layer. Finally, we added one dense layer with dropout layer followed by the output from the GRU layer. Then the second dense layer yielded the result. As the same as the CNN-LSTM model, we used SoftMax to predict the final multi-class result (Fig. 3). In this study, we shared the same CNN-GRU model structure for both MODIS and Sentinel-2 datasets.

2.5. Model implementation

One benefit of overlapping patches method is to increase the data volume for deep learning models. In the study, we split these small patches into two parts with 80% for training data and 20% for validating data. We tied up the feature images and label image as the input of the models.

All models were implemented through the Keras Python library with Tensorflow as the backend (https://keras.io/). This library is built on the top of the Tensorflow. We used Rectified Linear Unit (ReLU) activation function in the models, which is a powerful activation function in the deep learning models and this approach allows less computer calculation time and generates higher accuracy [34,45]. In addition, we used Categorical Cross Entropy as the loss function because of its ability to calculate the probability of each class [27,63]. To finalize, we made use of the Adam optimizer and set its learning rate to 0.0001. The Adam optimizer is a first-order stochastic gradient-based optimization algorithm to feedback the neural network, which is the most common and efficient optimizer in the deep learning models [64]. Considering the computer capability, the whole process was implemented on the A4 GPU with 48 GB storage and 768 GB RAM and trained models as 50 epochs. Notably, to get good classifiers, we set batch size as 16 for MODIS purpose, and 32 for Sentinel-2 purpose.

2.6 Model validation

In classifying remote sensing images, it is important to validate the model performance and evaluate classification results. During the training, we used the loss function to monitor the classifier models and adjusted the parameters. Then, we created confusion matrix to evaluate classification results from the test dataset. Specifically, we created a confusion matrix for each model to evaluate its overall accuracy, Precision (True Positive/ (True Positive + False

Positive)), Recall (Ture Positive/ (True Positive + False Negative)), and F1-score (2*(precision*recall)/ (precision + recall)).

3 Results

3.1. Training data and Model performance

We totally got 73,593 patches (58,874 for training and 14,719 for validate) for MODIS purpose, and 128,913 patches (103,128 for training and 25,785 for validate) for Sentinel-2 purpose. Since MODIS data has coarse spatial resolution, we created small patches by moving each one pixel along height and width of the image to make sure the models have enough data volume. However, because Sentinel-2 data has much finer spatial resolution, considering computing capability, we just created small patches by moving each 15 pixels along height and width dimensions. The total number of small patches was still double than MODIS data (Table 4). Because we used Sentinel-2 tile area as training data sites, we can clearly see both MODIS and Sentinel-2 data has the largest number of savanna class, which reminded that the study area is a savanna environment. Although Sentinel-2 data has finer spatial resolution than MODIS data, the number of cropland and grassland was similar with the MODIS data (Table 4). The explanation was that we used SCL from L2A product to mask out No data, cloud shadow, cloud medium probability, cloud high probability, thin cirrus, and snow or ice pixels and these unusable pixels could black out some cropland or grassland pixels. As we expected, both data has lowest number of forest class.

Table 4 Training small patch numbers for each class with MODIS and Sentinel-2 datasets

Classes	MODIS-train (80%)-validate (20%)	Sentinel-2-train (80%)-validate (20%)
Cropland	19,389	12,636
Grassland	20,034	27,736
Savanna	33,030	82670
Forest	1140	5871

To successfully classify different classes, the classifier is critical when considering deep learning methods. We used the same dataset to train three deep learning models to avoid bias came from the dataset itself. After evaluated validation data with trained models, the accuracy from the MODIS dataset was 94.62% for the CNN model, 96.25% for the CNN-GRU model, and 95.44% for the CNN-LSTM model. Sentinel-2 dataset, however, was 87.37% for the CNN model, 91.62% for the CNN-GRU model, and 91.53% for the CNN-LSTM model. The model results showed that features selected in this work was reasonable that optical bands and SWIR-1 band were useful for vegetation identification.

3.2. test data analysis

With the test site, we created confusion matrix using model results and ground truth data (Table 5 and Table 6) to quantitively evaluate model performance. The overall accuracy of the CNN-LSTM model (91.29% for MODIS and 89.47% for Sentinel-2) was higher than classification result from the CNN model (89.17% for MODIS and 86.02% for Sentinel-2) and the CNN-GRU model (89.19% for MODIS and 88.61% for Sentinel-2) for both datasets. The results determined that all deep learning models were robust for LULC classification problem, and the CNN-LSTM model had better overall performance than the CNN model and the CNN-GRU model in this research. Compared with the CNN model, the CNN-GRU and the CNN-LSTM models learned the temporal information of the pixel, which improved the classification accuracy. Although the CNN-GRU model had better overall accuracy than the CNN model, its accuracy was still lower than the CNN-LSTM model. Some studies also concluded that CNN-LSTM model had better performance than the CNN-GRU model in the classification task [62]. Furthermore, F1-score from our results also supported the points (Table 5 and Table 6). For example, MODIS data F1-score of cropland (0.974), grassland (0.288), savanna (0.952), and forest (0.528) from the CNN-LSTM was better than the CNN model and the CNN-GRU model, excepted the F1-score of cropland was 0.980 from the CNN-GRU model. Similarly, the Sentinel-2 data F1-score of cropland (0.942), grassland (0.390), savanna (0.933), forest (0.625) from the CNN-LSTM model was better than the other two deep learning models, excepted the F1-score of grassland was 0.405 from the CNN-GRU model. When investigated results from MODIS data and Sentinel-2 data, we found that F1-socre from grassland and forest in Sentinel-2 data had higher values than MODIS data. The possible reason could be that the pixel numbers in the Sentinel-2 test data were much larger than the MODIS data because of the finer spatial resolution with the same extent and the model had more training data to train.

Table 5 The confusion matrix of the classification results and ground truth data with all models from MODSI data

		cropland	grassland	savanna	forestland	precision	recall	F1-score	Overall Accuracy
CNN	cropland	670	4	0	0	0.9477	0.9941	0.970	0.8917
	grassland	3	286	131	0	0.1245	0.6810	0.210	
	savanna	34	2,008	20,318	105	0.9778	0.9044	0.940	
	forest	0	0	331	258	0.7107	0.4380	0.542	

Model		cropland	grassland	savanna	forestland	precision	recall	F1-score	Overall Accuracy
CNN-GRU	cropland	673	1	0	0	0.9628	0.9985	0.980	0.8919
	grassland	4	247	169	0	0.1816	0.5881	0.278	
	savanna	22	1,112	20,155	1,176	0.9856	0.8972	0.939	
	forest	0	0	126	463	0.2825	0.7861	0.416	
		cropland	grassland	savanna	forestland	precision	recall	F1-score	Overall Accuracy
CNN-LSTM	cropland	644	30	0	0	0.9923	0.9555	0.974	0.9129
	grassland	1	229	190	0	0.1954	0.5452	0.288	
	savanna	4	913	20,633	915	0.9885	0.9185	0.952	
	forestland	0	0	50	539	0.3707	0.9151	0.528	

Table 6 The confusion matrix of the classification results and ground truth data with all models from Sentinel-2 data

Model		cropland	grassland	savanna	forestland	precision	recall	F1-score	Overall Accuracy
CNN	cropland	1,560,655	98,111	77861	140	0.9259	0.8986	0.912	0.8602
	grassland	2,286	135,287	367,613	367	0.3712	0.2676	0.311	
	savanna	122,658	130,965	4,046,549	104,366	0.8918	0.9187	0.905	
	forest	28	121	45,727	103,507	0.4967	0.6929	0.579	
		cropland	grassland	savanna	forestland	precision	recall	F1-score	Overall Accuracy
CNN-GRU	cropland	1,609,861	94,809	31,830	267	0.9327	0.9269	0.930	0.8861
	grassland	45,980	185,890	272,639	1044	0.4501	0.3677	0.405	
	savanna	69,990	131,606	4,152,273	50,669	0.9164	0.9427	0.929	
	forest	274	704	74435	73,970	0.5873	0.4952	0.537	
		cropland	grassland	savanna	forestland	precision	recall	F1-score	Overall Accuracy
CNN-LSTM	cropland	1,631,676	64,246	40,726	119	0.9437	0.9395	0.942	0.8947
	grassland	46,310	167,863	290,794	586	0.4739	0.3320	0.390	
	savanna	49,718	121,771	4,196,063	36,986	0.9141	0.9527	0.933	
	forestland	1,263	372	62,806	84,942	0.6927	0.5686	0.625	