remote sensing

MDPI

Article

Deep Seasonal Network for Remote Sensing Imagery Classification of Multi-Temporal Sentinel-2 Data

Keli Cheng and Grant J. Scott *

Department of Electrical Engineering and Computer Science, University of Missouri, Columbia, MO 65211, USA; kchkr@umsystem.edu
* Correspondence: scottgs@missouri.edu

Abstract: As a medium-resolution multi-temporal data source, Sentinel-2 data has the potential to match the performance of using very-high-resolution (VHR) images in deep learning applications. To fully leverage the multi-temporal nature of Sentinel-2 data, we introduce the Deep Seasonal Network (DeepSN). This composite architecture combines a pre-trained deep convolutional neural network (DCNN) for visual feature extraction with a long short-term memory (LSTM) model to capture temporal information and make classification predictions. We evaluate the effectiveness of DeepSN on a Maasai Boma classification task in the Tanzania region. The DeepSN takes a sequence of four seasonal data, each spanning three months, for Boma prediction. Through cross-season validation experiments, we compare various advanced DCNNs and select EfficientNet as the backbone for DeepSN, as it performs the best. DeepSN with an EfficientNet backbone achieves a significant 19% improvement in the F1 score compared to plain EfficientNet for the Boma classification task. This work introduces a versatile composite architecture capable of handling multi-temporal data efficiently, providing flexibility in choosing the most suitable feature extraction backbone. The performance of DeepSN demonstrates the viability of utilizing medium-resolution multi-temporal data instead of high-resolution images for diverse tasks.

Keywords: composite neural network; mapping; multi-temporal neural network; Sentinel-2

check for updates

Citation: Cheng, K.; Scott, G.J.; Deep Seasonal Network for Remote Sensing Imagery Classification of Multi-Temporal Sentinel-2 Data. *Remote Sens.* **2023**, *15*, 4705. https://doi.org/10.3390/rs15194705

Academic Editor: Salah Bourennane

Received: 10 August 2023
Revised: 22 September 2023
Accepted: 24 September 2023
Published: 26 September 2023

Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

With effective aid provided by the United Nations Member States and international organizations, many underdeveloped countries have made significant progress toward achieving the Millennium Development Goals [1]. However, the minority populations and indigenous ethnic groups usually fail to benefit from wider improvements in health experienced by the general population in under-developed nations, e.g., Tanzania [2]. The Maasai are one of these marginalized minority populations, who reside in Tanzania and Kenya and are often described as typical pastoralists. The Maasai communities tend to be relatively remote, and pastoralism requires that most family units be at least semi-nomadic [2]. The semi-nomadic lifestyle of the Maasai people makes it challenging to locate their residential communities for public health records. As a result, many medicines and other living resources cannot be delivered to Maasai communities.

In order to connect the Maasai people to the existing public health services, the international non-profit organization, Humanity for Children (HFC) leads the Community Health Assessment Mapping Project (CHAMP) project to map the remote villages in Maasailand (Maasai Bomas). The initial Boma ground survey for the CHAMP project is accomplished by staff on motorcycles. Compared to the time-consuming and labor-intensive ground survey, combining remote sensing data and advanced machine learning techniques has the potential to be a drastically more effective approach to locating the Bomas and delivering health services. In [3], four advanced deep neural networks were evaluated using very high-resolution (VHR) satellite images available from the Google

Maps API. It was found that the best-performing architecture, ProxylessNAS [4], could be utilized for broad area scanning in [5]. While the use of VHR imagery enables the model to achieve a high accuracy rate, training the model and scanning become time-consuming as a result. Additionally, access to VHR typically has a cost that may be prohibitive for non-profit entities. In [6], multi-spectral medium-resolution Sentinel-2 data were evaluated as a possible alternative for Google Maps VHR imagery. However, the performance gap of models using these two types of data remains wide, despite the enhanced spectral fidelity and diversity of data bands available from Sentinel-2 data. In essence, the richer spectral data did not overcome the less exquisite spatial information. Therefore, the primary objective of this study is to explore how advanced deep neural networks (DNNs) may be used to fully exploit the multi-temporal feature of Sentinel-2 data, thereby closing the performance gap.

DNNs have seen explosive growth in computer vision applications across a wide swath of domains, and remote sensing is no different. In [7], in-depth techniques are provided to specialize the training and data augmentation for optimizing DNN performance on VHR remote sensing imagery. Transfer learning further facilitates the application of DNNs in areas with fewer data: for example, medical and remote sensing. In remote sensing, transfer learning has been widely applied to land-cover classification and object detection tasks [7–10]. Transfer learning has also been widely applied to assist semantic segmentation tasks. For example, ref. [11] utilizes transfer learning by pre-training the encoder layer of a semantic segmentation model on a large remote sensing dataset to learn generic features relevant to the domain. Additionally, VHR remote sensing imagery analytics have been tackled with composite DNN [12,13] as well as ensembles [14–17]. Remote sensing researchers have also pushed visual processing with deep learning beyond the classical *red-blue-green* channels into the full multi-spectral imagery (MSI) domain, e.g., [18–21].

With their capacity to extract intricate patterns and representations from data, DNNs have seen a diverse set of applications in remote sensing. In [22], a Siamese-based spatial-temporal attention architecture is designed for remote sensing change detection. Gargees and Scott, in [23], used features from transfer-learned DNN to perform land-cover analysis. Yang et al. in [24] proposed the deep heterogeneous superpixel network for remote sensing object localization. Other applications include but are not limited to wildfire detection [25], urban water extraction [26], crop yield prediction [27], and flood forecasting [28].

In this study, we explore the impact of incorporating temporal information to close the gap between VHR images and medium-resolution dataset performance. We propose a *Deep Seasonal Network*, *DeepSN*, to incorporate temporal information by way of a seasonal image sequence input into this new model design. The investigation is carried out through experiments with the DeepSN, which is a composite network, using both *deep convolutional neural network* (DCNN) components and *long short-term memory* (LSTM) recurrent neural components, that fuses the temporal information and spectral features of multi-temporal Sentinel-2 data. Multiple advanced DCNNs are evaluated for their effectiveness as the backbone of DeepSN. Experimental results demonstrate that DeepSN with the EfficientNet backbone is able to achieve comparable results on medium-resolution multi-temporal data to models using VHR data.

The remainder of this paper is organized as follows. Section 2 discusses the related prior research and state-of-the-art relevant to our proposed method. Section 3 details the datasets used, including location data, season alignment, data acquisition, and data processing methods. Section 4 describes the cross-season validation experiment used in this work and the proposed deep seasonal network. The experimental results are presented in Section 5 and discussed in Section 6. Finally, concluding remarks are provided in Section 7.

2. Related Work

Advanced methods of two main branches of neural machine learning are used in this study: the deep convolutional neural network and the recurrent neural network (RNN). Specifically, DCNN architectures are used to extract spectral and structural information

from Sentinel-2 imagery. An RNN architecture is used to process the visual features from multi-temporal datasets as sequences. In this work, we build off the specific styles of these types of neural networks, constructing a composite DNN to achieve enhanced classification performance for the task of classifying multi-temporal Sentinel-2 image chips that have Boma within them.

2.1. DCNN

A convolutional neural network (CNN) is a type of deep learning neural network that is commonly used for image and video classification recognition tasks. A CNN typically consists of convolutional layers, pooling layers, and fully connected layers. The convolutional layers apply a set of learnable filters to the input and scan over it to extract features from different locations. Each of the filters is a small matrix that is applied to a small region of the input (receptive field). The filter slides over the input, element-wise multiplying and summing the values in the receptive field with the filter weights, producing a new output value. A set of feature maps representing different features of the input are generated by repeating the filtering process. Since the parameters of the filters are learned through training, the feature maps generated through the convolutional operations are the most relevant features for the task. DCNN is a type of CNN with deep depth, typically consisting of more than ten convolutional layers. The deep architecture allows DCNNs to learn more complex and abstract features from the inputs, where each layer is assembling lower-level structures into higher-level visual concepts. Eventually, the deep convolutional features are passed into a classification or regression phase within the DCNN for the ultimate task output. Examples of DCNNs include VGGNet [29], ResNet [30], Inception [31], DenseNet [32], and EfficientNet [33], just to list a few widely used architectures. In this study, several advanced DCNNs are evaluated for their ability to extract features from multi-temporal datasets.

2.2. RNN

The second key neural machine learning architecture related to this work is an advanced RNN that characterizes the sequential property of multi-temporal feature vectors for the Boma classification task. RNN is a type of neural network that is designed to handle sequential data, such as time series, speech, and text. The key difference between RNNs and traditional feed-forward neural networks is that RNNs have hidden states that function as "memory" to maintain the information from previous time steps. The hidden state is passed from one time-step to the next, allowing the network to maintain information about previous steps, as shown in Figure 1. These recurrent characteristics make RNN useful for difficult tasks involving sequential data analysis, such as language processing [34], machine translation [35], and speech recognition [36].

In this study, we use the LSTM model, an advanced RNN architecture, to learn the seasonal transition pattern under the multi-temporal dataset. LSTM is an advanced RNN that is designed to address some of the shortcomings of regular RNNs. One major shortcoming of regular RNNs is that the gradient vanishes through training. As the gradient becomes smaller, it becomes harder for the network to update weights and takes longer to achieve the optimal result. The problem is solved by the gates and the cell state introduced in LSTM architecture. The authors recommend an in-depth examination of the LSTM model in [37] for interested readers.

2.3. CNN & RNN Composites

In general, a composite architecture combines the strengths of different types of neural networks to create a more powerful and versatile model. One example of such a composite architecture is ConvLSTM, a type of RNN for spatiotemporal prediction that has a convolutional structure in both the input-to-state and state-to-state transitions [38]. ConvLSTM has been applied to various tasks; for instance, ref. [39] uses it to extract spatial-spectral features for hyperspectral image classification. Instead of having a convolutional

structure in the LSTM state transitions, another type of composite architecture is to have the CNNs extract spectral features first, and then use RNNs to process the features to capture the temporal information. One advantage of this approach is that one can use advanced and pre-trained CNNs as feature extractors. For example, Ref. [12] proposes Re-ResNet that combines the LSTM architecture with ResNet50 architecture for an urban land cover classification task. There are many other applications of composite architectures in the field of remote sensing, such as [13,40]. In [13], a deep Siamese convolutional multi-layer RNN for the change detection task through VHR images is leveraged. In [40], a unified architecture using 3-D CNN and bidirectional LSTM is proposed for hyperspectral image classification. The joint spatio-temporal learning through CNN-RNN composites allows better exploitation of information from sequential remote sensing imagery.

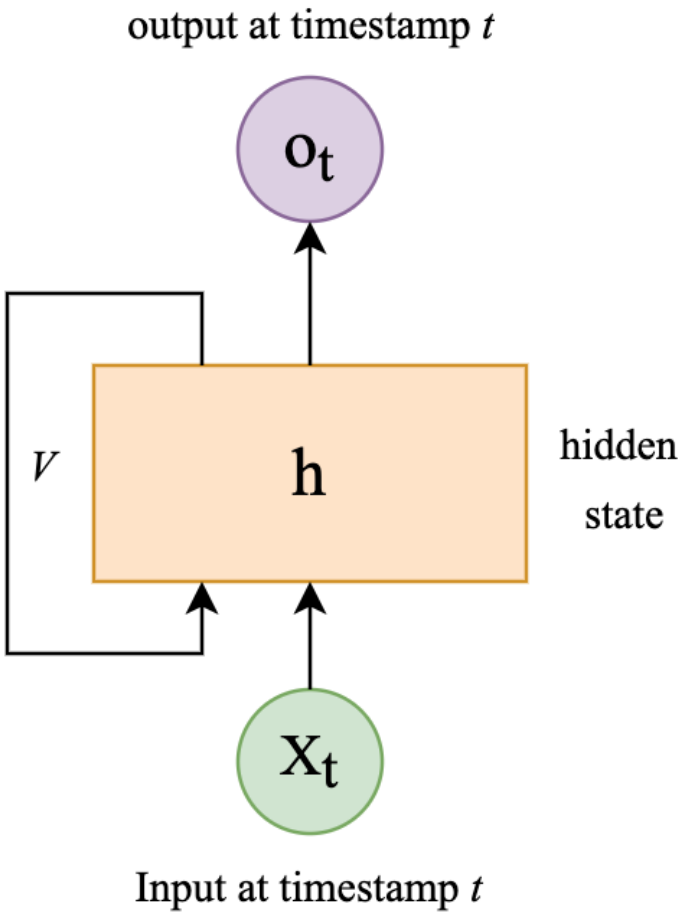


Figure 1. Demonstration of the data flow in recurrent neural networks. The hidden state is passed from one time-step to the next, allowing the network to maintain information about previous steps.

3. Data Collection

3.1. Boma Locations

The study area of interest (AOI) in this work spans 3906 km^2 between latitude and longitude measurements of 3°58′37″S, 36°09′05″E and 4°31′18″S, 36°43′53″E. Including, but not limited to, grasslands, wetlands, and mountains, the AOI spans the administrative regions of Manyara, Dodoma, Arusha, and Singida.

To locate Bomas, a manual survey of the AOI was carried out using Google Earth Pro. The visual characteristics that distinguish Bomas and the other constructions are the circular shape and their unique corral fencing structures, as shown in Figure 2. Based on these two criteria, 635 positive Boma samples were collected through the survey. Negative samples' (Non-Boma) coordinates were automatically created with a 1 km offset in four arbitrarily

chosen cardinal and inter-cardinal directions. The georeferenced satellite images were then cut into individual image chips using the manually gathered Boma and Non-Boma coordinates. The image chips of Non-Boma samples were further refined to remove those with partial Boma structures that appeared. The total number of Non-Boma samples was 1726 after refinement.

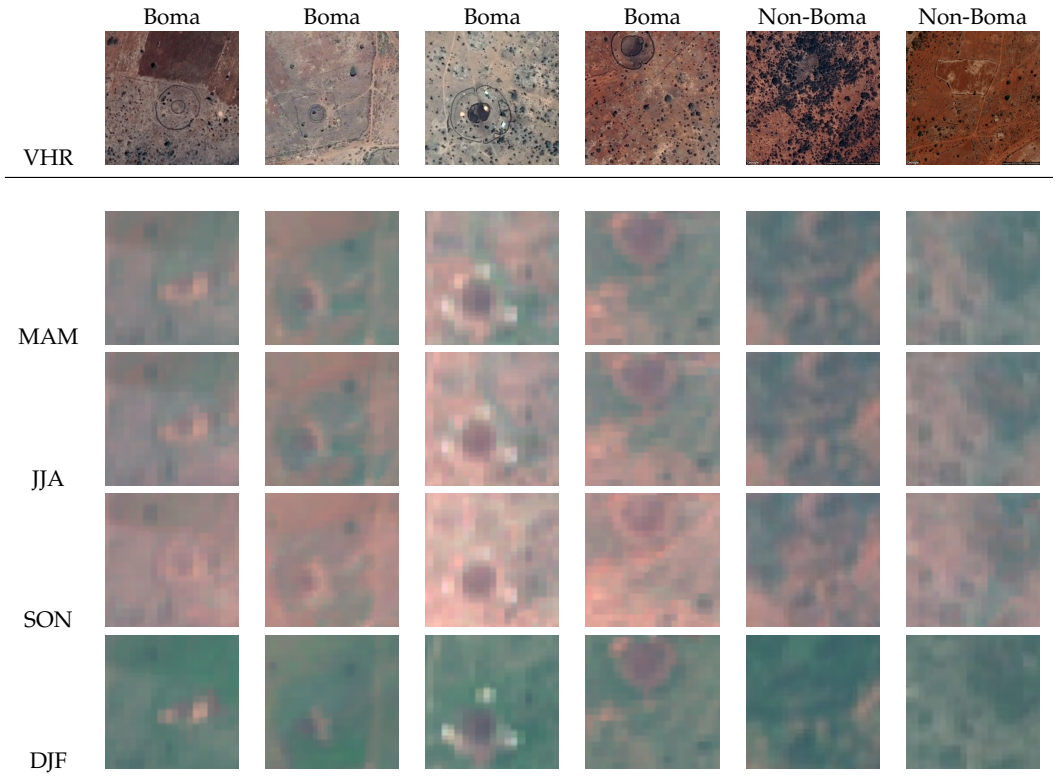


Figure 2. Example very high-resolution (VHR) and Sentinel-2 images of Bomas (first 4 columns) and Non-Bomas (last 2 columns) collected during different aligned seasons. At a similar zoom level, most of the Boma structures inside the VHR images are distinct and easy to be recognized, in contrast to the considerably blurrier Sentinel-2 images. However, Bomas have different topographical features during different seasons, making them more recognizable during certain seasons. The multi-temporal Sentinel-2 data is able to capture seasonal characteristics and transitions.

3.2. Sentinel-2 Data

In order to investigate the effect of multi-temporal data on image classification, we chose the Sentinel-2 data for experiments. The Sentinel-2 data are the product of the Copernicus Sentinel-2 mission, a multi-spectral and broad-swath imaging mission.

In comparison to VHR images, the data resolution of the Sentinel-2 data is low, but its resolution is still suitable for the visual analysis of structural occurrences of land cover, such as Boma detections. Therefore, it is generally considered a medium-resolution data source. Although the resolution of Sentinel-2 data is lower than that of VHR imagery, it has a higher revisit frequency: approximately every 5 days for any given location on Earth. With the high revisit frequency, the Sentinel-2 data have been widely used for land monitoring and other projects that use sequential data.

The dataset for this study was produced using the Sentinel-2 level 1C product that was obtained using Google Earth Engine (GEE). The imagery of the AOI was collected during the year 2020 at scale 3 (3.004 m/px), with less than 20% cloud pixel percentage. We used the median of each seasonal dataset as the representative source data and used GEE's visualization tool to convert the original 16-bit data into 8-bit RGB data. In the presence of outliers (such as clouds and shadows), the median function is necessary to remove the high-intensity (clouds) and low-intensity (shadows) in order to achieve a representative

result that is not biased in the form of an average. The normalized imagery is further cropped into 64 × 64 px chips according to the locations of Boma and Non-Boma examples. Each of the Boma and Non-Boma data points is associated with a sequence of four seasonal image chips that span three months equally, as shown in Figure 2.

3.3. Season Alignment

The Maasai reside in a semi-arid ecology prone to erratic rainfall and periodic drought [2]. Most of the Tanzania area experiences two rainy seasons and two dry seasons every year. The short dry and short rain seasons last about two months, while the long dry and long rain seasons last three to five months [41].

There is a problem with clouds if the seasonal data are fully generated based on the seasonal characteristics in Tanzania. Especially for the datasets that are collected during the short rain season, the shorter time span and the greater number of clouds become a challenge for data processing. To address this problem, we aligned the seasons evenly so that each season spanned three months, i.e., quarter years. Figure 3 shows the periods of the original rainy dry season and our adjusted seasons.



Figure 3. Season alignment of rain and dry seasons to selected 3-month quarters. The 3-month quarters are chosen to evenly divide the annual data collections.

In Figures 2 and 3, the following 3-month indicators are used: *MAM* for March, April, May; *JJA* for June, July, August; *SON* for September, October, November; and *DJF* for December, January, February.

Figure 2 displays the VHR images of Bomas and Sentinel-2 images of the same locations that are collected during different seasons. At a similar zoom level, most of the Boma structures inside the VHR images are distinct and easy to recognize, while those on the Sentinel-2 images are pixelated and blurry. The examples selected here are images of medium to large-size Boma, thus they are still recognizable on the Sentinel-2 images. There are smaller and less recognizable Boma samples that exist in the dataset. Although the resolution of Sentinel-2 images is much lower than that of the VHR images, they have some advantages: for Bomas that are surrounded by vegetation, their structures may become more distinct on Sentinel-2 images that are collected during arid seasons. Moreover, there is a clear shift of the landform features between seasons, as depicted in the images. Learning the landform transition pattern from the multi-seasonal images may be helpful for Boma identification.

4. Method

4.1. DCNN as Feature Extractor

In this study, we evaluate several DCNN architectures for their generalizability on the multi-seasonal dataset: the ResNet50, ProxylessNAS, and EfficientNet.

ResNet-50 is a residual network [30] with 50 convolutional steps. It relies on micro-architecture modules, instead of traditional sequential network architectures. Based on its simple structure, it is often used as a baseline with which to compare other more advanced and complex models.

We also evaluate the ProxylessNAS architecture because of its leading performance in Boma mapping tasks in [5]. ProxylessNAS is a variant of Neural Architecture Search (NAS), which directly optimizes neural network architectures on the target task [4]. It is demonstrated to efficiently expand the search space and achieve greater performance than previous proxy-based NAS algorithms since it searches without any proxy while still enabling a large candidate set and removing the restriction of repeating blocks.

Compared to ResNet-50 and ProxylessNAS, EfficientNet typically performs better on a variety of tasks. It employs a straightforward yet efficient compound coefficient to scale up convolutional networks in width, depth, and resolution. The architecture is developed through a multi-objective neural architecture search that optimizes both the accuracy and the compound coefficient [33]. In the experiments, we use the EfficientNet-B0 architecture, which is the EfficientNet baseline. The EfficientNet-B0 has lower computational requirements than other EfficientNet variants while still having higher accuracy than ResNet-50 and many other architectures when tested on the ImageNet dataset.

In the experiments, we use models that are pre-trained on the ImageNet dataset and fine-tune them on the seasonal Boma dataset, i.e., transfer learning the convolutional feature extractor.

4.2. Cross-Season Validation

To evaluate the generalizability of each model on seasonal datasets, we conduct the cross-season validation experiments. As shown in Figure 2, similar structural characteristics are shared by the seasonal chips from the same area. However, the spectral characteristics change with the seasons. Therefore, cross-season validation is an effective means of investigating what the model has learned from structural and spectral features. Additionally, using one model that generalizes well across seasonal datasets eliminates the need for four season-specific feature extractors, which can significantly lower the computational cost and model complexity. We fine-tune the models using the imagery from one season and validate their performance using the other three seasons. Figure 4 lists the training and validation datasets used in the cross-season validation experiments.

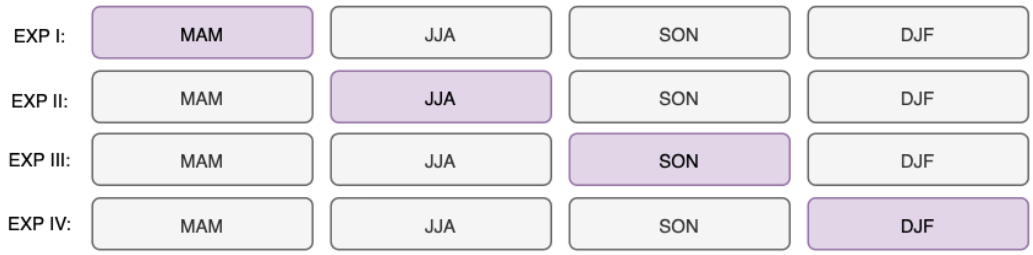


Figure 4. Cross-season validation datasets: For each set of experiments, the purple data serve as training data and the gray data serve as validation data.

4.3. Deep Seasonal Network

In this work, we propose the DeepSN to fuse the visual features and temporal information of a multi-seasonal dataset. The DeepSN uses a fine-tuned DCNN to extract visual features from the multi-temporal dataset and feed the features into the LSTM architecture to fuse the temporal information and perform the final prediction.

Figure 5 visualizes the proposed architecture. Each input data point for the architecture consists of four seasonal images that are collected during MAM, JJA, SON, and DJF seasons, respectively. We use a pre-trained EfficientNet-B0 as the feature extractor to generate $1280 \times 7 \times 7$ feature maps from the input images. A max-pooling layer is then applied on the feature maps to flatten them into 1280×1 feature vectors and a dense layer is applied to further reduce the dimension of input vectors to 64 per season. The feature vectors extracted from seasonal datasets are connected through a 1-layer LSTM to learn the transition pattern between the seasonal vectors.

One limitation of traditional RNNs is that they treat every component of the sequence equally. However, in some forecasting problems, the closer nodes typically have a greater impact on forecasting. As a result, it is necessary to overlook some prior information and move forward. The forget gate in LSTM determines which information is to be omitted through Equation (1). The output of forget gate f_t ranges between [0.0, 1.0] for each

number in the cell state c_{t-1}, where 1.0 represents completely keeping the information (remembering) and 0.0 represents completely omitting it (forgetting).

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t]). \tag{1}$$

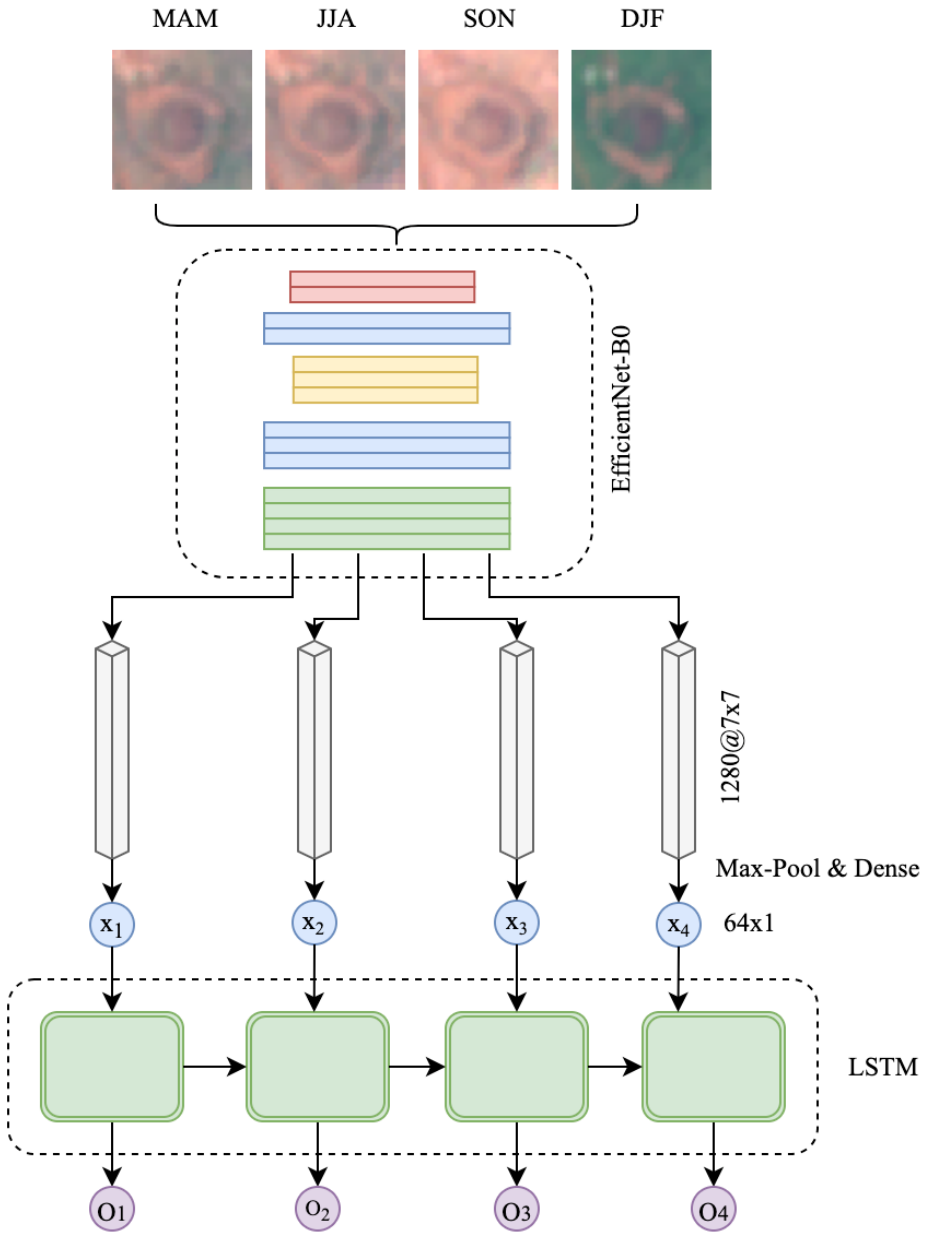


Figure 5. Deep seasonal network: a model that uses pre-trained deep convolutional neural network (DCNN) to extract visual features from a multi-seasonal dataset and uses the long short-term memory (LSTM) layer to learn the transition pattern between seasonal vectors to make predictions.

Similarly, the input gate determines what new information is to be stored in the cell state. The input is updated using Equation (2). The *sigmoid* function, σ, is used to decide which values to update and the *tanh* function is used to create a new candidate vector $\tilde{c}_t$ based on the hidden state from previous timestamp h_{t-1} and new current information x_t.

$$\begin{aligned} i_t &= \sigma(W_i \cdot [h_{t-1}, x_t]) \\ \tilde{c}_t &= \tanh(W_c \cdot [h_{t-1}, x_t]). \end{aligned} \tag{2}$$

In terms of memorization, LSTM introduces the cell state to encode long-term dependencies and relations. The cell state allows information to flow through the entire LSTM architecture. The outputs from the forget gate, input gate, and the candidate vector are used to update the cell state using Equation (3).

$$c_t = c_{t-1} \otimes f_t \oplus (\tilde{c_i}) \otimes i_t. \tag{3}$$

The output gate regulates the data delivered to the network as input for the next phase from the data encoded in the cell state. The hidden state h_t and output o_t are computed through Equation (4).

$$\begin{aligned} o_t &= \sigma(W_o \cdot [h_{t-1}, x_t]) \\ h_t &= o_t \otimes \tanh(c_t). \end{aligned} \tag{4}$$

In the calculation of the loss, we accumulate the loss from all four outputs and compute the cross-entropy loss, as shown in Equation (5), where y_i is the true label of given sample.

$$L(\theta) = -\sum_{i=1}^{4} y_i \log(o_i). \tag{5}$$

5. Experiments

Two primary sets of experiments were undertaken: the evaluation of different DNN backbones and the assessment of the proposed DeepSN model. In the DNN backbone evaluation (Section 5.1), we investigated the backbones' ability to generalize across seasonal datasets. The investigation involves the evaluation of advanced DNN architectures through cross-season validation experiments. Specifically, we partitioned the data, utilizing one season for training and the remaining seasons for validation. In Section 5.2, we assess the DeepSN through the classical five-fold cross-validation experiments on the multi-temporal dataset constructed as described in Section 3, where each data point was associated with four seasonal image chips. Within each fold of the cross-validation, 80% of the multi-temporal data were designated for training, while the remaining 20% were reserved for testing.

5.1. Cross-Season Validation

Three DCNN architectures are evaluated for cross-season validation: the ResNet50, ProxylessNAS, and EfficientNet-B0. The goal is to see which DCNN will provide the best feature extraction backbone in the DeepSN. In these experiments, data from only one season, such as MAM, are used during training, and the remaining three seasons, e.g., JJA, SON, and DJF, are used to evaluate the trained model. This is effectively 25% training, 75% testing rotating data splits. The training hyperparameters for the DCNNs include 64 batch size, the Adam optimizer, and an initial learning rate of 1×10^{-3} with the cosine annealing scheduler. The models are trained for 80 epochs with expanding augmentations of a random choice of flip or rotation. The augmentation operations make it equivalent to training for 160 epochs.

Table 1 shows the results of cross-season validation experiments. In general, models trained with MAM data have a leading performance when validated on data from other seasons. The reason behind this may be that the chips generated from MAM data are mixed with images of an arid and wet season. According to [2], the rainfall where the Maasai live is erratic, even during the rainy season.

In terms of DCNN models, the EfficientNet has the best performance, while the ProxylessNAS has the lowest performance. The EfficientNet trained on MAM data reaches an average F1 of 75.90% when testing on JJA, SON, and DJF data. Therefore, EfficientNet is chosen as the feature extractor for the DeepSN.

Table 1. Cross-season validation results. Top results in bold for each metric.

Architecture	Training Season	Precision	Recall	F1
ResNet50	MAM	85.70%	59.51%	70.24%
	JJA	86.16%	53.03%	65.65%
	SON	79.79%	43.31%	56.15%
	DJF	75.48%	24.45%	36.94%
ProxylessNAS	MAM	65.25%	47.28%	54.83%
	JJA	70.07%	44.78%	54.64%
	SON	63.19%	28.79%	39.55%
	DJF	53.40%	13.11%	21.06%
EfficientNet	MAM	94.61%	**63.38**%	**75.90**%
	JJA	**95.28**%	60.08%	73.70%
	SON	93.33%	51.10 %	66.04%
	DJF	90.53%	21.06 %	34.17%

5.2. Performance of Deep Seasonal Network

The results from cross-season validation experiments show that the EfficientNet exhibits the best generalizability across the multi-temporal dataset, thus we use it as the feature extractor to generate representative features from the dataset. We have run five-fold cross-validation experiments on the DeepSN to evaluate its performance with different pre-trained feature extractors. For each fold, the ratio of positive to negative samples is identical to the ratio for the entire dataset. Per fold, there are approximately 127 Boma and 345 Non-Boma samples. The training parameters for the Deep Seasonal Network include a batch size of 16, Adam optimizer with 10^{-4} initial learning rate, and 80 training epochs. Note, the batch size of 16 in the multi-temporal, e.g., 4 seasons, data are equivalent to a 64 batch size in the experiments of Section 5.1.

Table 2 shows the performance of DeepSN with feature extractors that were pre-trained on the different seasonal datasets. We also compare the performance of the DeepSNs to EfficientNet architectures that are trained and tested using identical settings with the DeepSN. The experiment is to emphasize the improvement led by the newly introduced LSTM component and intermediate layers in DeepSN. As desired, the DeepSN is able to exploit the multi-temporal data and results in a leading performance, an F1 of 99.92%, with the feature extractor that is trained on the MAM dataset. Regardless of the training data, DeepSN significantly improves performance over EfficientNet alone in general. Table 2 demonstrates that the DeepSN outperforms EfficientNet in terms of F1 error reduction, with a range of 23.02% to 99.60%.

Table 3 compares the performance of related architectures on the Boma classification task evaluated through five-fold cross-validation experiments. The DeepSN with the EfficientNet backbone is able to attain an F1 score of 99.92%, surpassing the highest F1 score of 97.28% reported in [5] using VHR imagery. DeepSN provides a more noticeable improvement with the same resolution data. In the absence of DeepSN, the ProxylessNAS achieves an F1 score of 73.42% when utilizing annual median data from Sentinel-2 [6]. Nevertheless, when incorporating multi-temporal Sentinel-2 data in conjunction with DeepSN, the F1 score experiences a substantial improvement, reaching 92.12%. Likewise, when employing the EfficentNet backbone, the enhancement in performance is also notable, increasing the F1 score from 73.42% to 99.92%. The comparison reveals that DeepSN not only maximizes the utilization of Sentinel-2 data but also attains an equivalent level of performance on the Boma classification task as using VHR images.

Table 2. Performance of deep seasonal network with feature extractors that are trained on different seasonal datasets. The performance was compared to EfficientNet alone using the same training and testing settings. Top results in bold for each metric.

Train Data	Model	Precision	Recall	F1	F1 Error Reduction
MAM	EfficientNet	90.82%	72.53%	80.63%	–
	DeepSN	**99.92**%	**99.92**%	**99.92**%	99.60%
JJA	EfficientNet	92.08%	70.06%	79.53%	–
	DeepSN	96.40%	94.40%	95.39%	77.46%
SON	EfficientNet	89.51%	63.32%	74.17%	–
	DeepSN	90.69%	83.19%	86.78%	48.82%
DJF	EfficientNet	90.10%	40.64%	55.95%	–
	DeepSN	78.46%	57.09%	66.09%	23.02%

Table 3. Comparison of five-fold cross-validation experimental results using other data source and models. Top results in bold for each metric.

Architecture	Data Source	Precision	Recall	F1
ProxylessNAS	VHR	97.38%	97.29%	97.29%
ProxylessNAS	Sentinel-2 1C Annual Median	73.15%	73.90%	73.42%
EfficientNet	Sentinel-2 1C Annual Median	73.05%	74.04%	73.42%
DeepSN-ProxylessNAS	Sentinel-2 1C Multi-Temporal	95.62%	88.87%	92.12%
DeepSN-EfficientNet	Sentinel-2 1C Multi-Temporal	**99.92%**	**99.92%**	**99.92%**
EfficientNet (MAM-trained)	Sentinel-2 1C Multi-Temporal	90.82%	72.53%	80.63%

The results using ProxylessNAS are copied from [5,6].

5.3. Evaluating Robustness by Withholding a Season from Inference

The feature extractor in DeepSN is trained using one of the seasonal datasets, making the generalizability of the whole system questionable in terms of the locked feature extractor being encoded with data that may be in the testing domain. In order to eliminate the possible effect of the testing data used to train the feature extractor, we remove the MAM images from the validation datasets and repeat the experiment on the MAM-trained feature extractor used in the DeepSN. To be clear, we are purposefully degrading the input data and the DeepSN system, as the testing data now has only three seasons instead of all four during evaluation. Table 4 presents the results of this experiment. Although the F1 score of using MAM-trained DeepSN decreased from 99.92% to 85.71%, it is still significantly higher than using EfficientNet alone (73.20%). In comparison with EfficientNet, the DeepSN still leads to a 46.68% error reduction on F1 when evaluated with only a partial year of data.

We also provide the scores with an enhanced evaluation scheme in Table 4. In terms of accuracy, three different evaluation schemes are considered: ACC_{all}, ACC_{last}, and ACC_{any}. ACC_{all} uses all the output labels to compute the confusion matrices. It provides a more intuitive comparison between the regular EfficientNet and DeepSN. ACC_{last} follows the "many-to-one" design model of RNN, a commonly used model for classification problems. With the ACC_{last} scheme, only the last output o_4 is considered as the final prediction. ACC_{any} is a special evaluation scheme for Boma classification, where we consider the whole sequence as Boma if any seasonal chip is categorized as Boma, such that $o = min(o_1, o_2, o_3, o_4)$. When comparing the EfficientNet-B0 to the degraded DeepSN, the precision and Recall increased with DeepSN ACC_{last} from 86.64% and 63.38%, respectively, to 91.15% and 82.29%, respectively; the F1 score increased as well from 73.20% to 86.49%. With DeepSN ACC_{any}, the Recall increased to 86.05% while the precision dropped to 87.70%, leading the F1 to increase to the highest 86.87%.

Table 4. Season degraded inference study. Top results in bold for each metric.

Architecture	Scheme	Precision	Recall	F1
EfficientNet-B0	–	86.64%	63.38%	73.20%
Deep Seasonal Net	ACC_{all}	89.25%	82.45%	85.71%
Deep Seasonal Net	ACC_{last}	**91.15**%	82.29%	86.49%
Deep Seasonal Net	ACC_{any}	87.70%	**86.05**%	**86.87**%

5.4. Efficiency

Within the DeepSN framework, the training and feature extraction consume the most significant amount of time. The training time for the EfficientNet backbone is approximately 45 min when using a Tesla T4 GPU. Table 5 provides a comparison of the training time for the DCNNs used in this study. It is evident from the table that models trained with Sentinel-2 data exhibit significantly shorter training time compared to those using VHR images. This is especially important, as the multi-temporal data within the DeepSN achieve comparable performance as a fraction of the training time.

Feature extraction requires approximately 45 min for the four seasonal datasets, encompassing both I/O operation time and model inference time. With the feature extractor frozen and feature vectors cached, the effective training time for DeepSN, using a 2-layer LSTM, is around 21 min, utilizing an NVIDIA GeForce GTX 1080 GPU. The training time will experience a slight increase if employing a more complex LSTM architecture, but it is not anticipated to be excessively time-consuming.

Table 5. Computation time comparison of DCNN models. The table lists the training time in hours for DCNN models involved in this study using VHR images and multi-temporal Sentinel-2 images.

Architecture	Data Source	Resolution	GPU	Time (h)
ResNet50	VHR	512 px	NVIDIA Tesla P100	8.57
ProxylessNAS	VHR	512 px	NVIDIA Tesla P100	6.92
ResNet50	Sentinel-2 1C	64 px	NVIDIA GeForce RTX 3090	0.45
ProxylessNAS	Sentinel-2 1C	64 px	NVIDIA GeForce RTX 3090	0.49
EfficientNet	Sentinel-2 1C	64 px	NVIDIA Tesla T4	0.75

6. Discussion

Here, we propose a composite architecture that takes advantage of advanced DCNN and RNN architectures. The performance of the proposed deep seasonal network is evaluated on the Boma classification task and achieved a promising performance. In this task, it should be noted that Recall is the most relevant metric, as it measures the capability of detecting Boma in broad area scans, which is the application use case. Specifically, the Recall of 99.92% for DeepSN with a MAM-trained feature extractor exceeds the classification of VHR-trained detectors on this data set [5].

We evaluate different DCNN architectures as feature extractors through a novel cross-season validation. The validation process measured how the model trained on one-seasonal dataset performs on the other seasonal datasets, and in this regard, the variability of the March, April, and May (MAM) training data for a feature extractor was found to perform the best. The possible reason behind this is that the data of this season possesses the characteristics of both rainy and dry seasons: the images show a mixture of vegetation and land. This leads to better generalizability of MAM-trained architecture. Both fine-tuning a DCNN that adapts to all the seasonal datasets and training season-specific feature extractors require voluminous data and excessive training time. The representative feature extractor can be easily discovered through the cross-season validation, thus reducing the number of feature extractors needed in the proposed architecture.

Among the three evaluated DCNNs, the EfficientNet achieves the highest Recall and F1 scores and was selected to be the feature extractor in DeepSN. It is discovered through

experiments that the feature extractor is the most fundamental element of the proposed architecture. Not only does it consume the most time to train and extract features, but it also determines how well the DeepSN performs as a whole. The LSTM component fuses the temporal information and, in general, will lead to more than 10% improvement in F1 scores (see Table 2). If the feature extractor is unable to characterize the target object or exhibits insufficient generalizability across the multi-temporal dataset, it significantly reduces the performance of the DeepSN. Given that a location's structural features remain constant and that spectral features are crucial for reflecting temporal changes, the DCNN's capacity to generate spectral features is essential for processing multi-temporal data.

We then evaluate the performance of DeepSN through comparisons with plain EfficientNet. The findings of the experiment demonstrate that the DeepSN significantly outperforms the basic EfficientNet. The degraded seasonal data study that removes MAM data from the validation dataset further validates the conclusion, showing that the DeepSN still performs significantly better than EfficientNet (see Table 4). Additionally, we look into several evaluation schemes, and the experimental findings reveal that ACC_{any} leads to the highest F1 score. The exploration of different evaluation schemes provides instructions for real-world applications of DeepSN.

As highlighted, the purpose of this study is to fully utilize the multi-temporal characteristics of Sentinel-2 data to close the performance gap with VHR data. In this work, we increase the input data dimension by constructing a four-season image sequence for each data point. However, the findings from Tables 2 and 4 demonstrate that extended input cannot result in improvement, utilizing EfficentNet alone. This disparity can be attributed to the limitation of EfficientNet, which solely focuses on capturing visual features and lacks the capability to learn the underlying seasonal transition patterns. In contrast, the proposed DeepSN architecture captures both visual and temporal features, resulting in substantial improvements. Moreover, DeepSN requires significantly less training time than models using VHR images. Overall, the proposed architecture fulfills the need for an accurate and efficient model to enable broad-area land cover mapping in the future using Sentinel-2 time series.

7. Conclusions

In this study, the DeepSN architecture for multi-temporal data is proposed, and its effectiveness is evaluated through the Maasai Boma classification task. This network is especially well-suited for the Boma use case, where the structures are non-permanent and therefore not well suited for multi-year approaches. We designed novel cross-season validation experiments to evaluate the generalizability of different DCNN backbones and the training data for the feature extractors. DeepSN with the EfficientNet backbone was compared with plain EfficientNet through five-fold cross-validation. With DeepSN, the Recall and F1 scores are significantly improved. We also conducted a follow-up experiment to eliminate the possible effect of the training data for feature extractors and prove the validity of the preceding experiments again. Additionally, various evaluation schemes are compared to provide useful insight.

This work presents a typical composite architecture for multi-temporal and multi-spectral data processing. This type of composite architecture ensures enough flexibility in the choice of backbone networks. The outstanding performance of DeepSN also shows the feasibility of replacing VHR images with medium-resolution multi-temporal data in various tasks.

There is still plenty of room to explore DeepSN. For instance, the complexity of LSTM architecture and the order of visual features being fed into LSTM. In general, LSTM is sensitive to the first sample it receives. Thus, the order of sequential data may play a role in the performance of DeepSN. Also, increasing the complexity of LSTM may improve performance but may also lead to higher computational costs. Furthermore, additional DCNN and vision transformer backbones may be evaluated for their visual feature extraction capability across different seasons. The key visual factor varies for different application tasks, and so does the appropriate feature extractor. Since Sentinel-2 is a multi-spectral data source with a high revisit frequency, both the length and the width of sequential data can

be extended to capture more spectral and temporal information. However, extending the temporal dimension of the data will significantly increase the computational cost.

Author Contributions: K.C.: methodology, software, validation, formal analysis, investigation, resources, data curation, writing original draft preparation, writing review and editing; visualization. G.J.S.: conceptualization, methodology, validation, resources, writing review and editing, supervision, project administration. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data presented in this study are openly available in https://github.com/MU-HPDI/Cheng_Boma_Data, (accessed on 9 August 2023).

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

References

1. Bank, A.D. A Critical Appraisal of the MDGs. MDGs in Tanzania: Progress and Challenges. 2009. Available online: https://sarpn.org/documents/d0001556/P1909-Afrodad_MDGs_Tanzania.pdf (accessed on 9 August 2023).
2. Lawson, D.W.; Borgerhoff Mulder, M.; Ghiselli, M.E.; Ngadaya, E.; Ngowi, B.; Mfinanga, S.G.; Hartwig, K.; James, S. Ethnicity and child health in northern Tanzania: Maasai pastoralists are disadvantaged compared to neighbouring ethnic groups. *PLoS ONE* **2014**, *9*, 0110447. [CrossRef] [PubMed]
3. Cheng, K.; Popescu, I.M.; Sheets, L.; Scott, G.J. Automatic Maasailand Boma Mapping with Deep Neural Networks. In Proceedings of the 2021 IEEE International Geoscience and Remote Sensing Symposium IGARSS, Brussels, Belgium, 11–16 July 2021; pp. 2839–2842. [CrossRef]
4. Cai, H.; Zhu, L.; Han, S. ProxylessNAS: Direct Neural Architecture Search on Target Task and Hardware. *arXiv* **2018**, arXiv:1812.00332.
5. Cheng, K.; Popescu, I.; Sheets, L.; Scott, G.J. Analysis of Deep Learning Techniques for Maasai Boma Mapping in Tanzania. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2022**, *15*, 3916–3924. [CrossRef]
6. Cheng, K.; Bajkowski, T.M.; Scott, G.J. Evaluation of Sentinel-2 Data for Automatic Maasai Boma Mapping. In Proceedings of the 2021 IEEE Applied Imagery Pattern Recognition Workshop (AIPR), Washington, DC, USA, 12–14 October, 2021; pp. 1–5. [CrossRef]
7. Scott, G.J.; England, M.R.; Starms, W.A.; Marcum, R.A.; Davis, C.H. Training Deep Convolutional Neural Networks for Land-Cover Classification of High-Resolution Imagery. *IEEE Geosci. Remote Sens. Lett.* **2017**, *14*, 549–553. [CrossRef]
8. Hu, F.; Xia, G.S.; Hu, J.; Zhang, L. Transferring Deep Convolutional Neural Networks for the Scene Classification of High-Resolution Remote Sensing Imagery. *Remote Sens.* **2015**, *7*, 14680–14707. [CrossRef]
9. Chen, Z.; Zhang, T.; Ouyang, C. End-to-End Airplane Detection Using Transfer Learning in Remote Sensing Images. *Remote Sens.* **2018**, *10*, 139. [CrossRef]
10. Fang, B.; Kou, R.; Pan, L.; Chen, P. Category-Sensitive Domain Adaptation for Land Cover Mapping in Aerial Scenes. *Remote Sens.* **2019**, *11*, 2631. [CrossRef]
11. Panboonyuen, T.; Jitkajornwanich, K.; Lawawirojwong, S.; Srestasathiern, P.; Vateekul, P. Semantic Segmentation on Remotely Sensed Images Using an Enhanced Global Convolutional Network with Channel Attention and Domain Specific Transfer Learning. *Remote Sens.* **2019**, *11*, 83. [CrossRef]
12. Qiu, C.; Mou, L.; Schmitt, M.; Zhu, X.X. Local climate zone-based urban land cover classification from multi-seasonal Sentinel-2 images with a recurrent residual network. *ISPRS J. Photogramm. Remote Sens.* **2019**, *154*, 151–162. [CrossRef]
13. Chen, H.; Wu, C.; Du, B.; Zhang, L.; Wang, L. Change Detection in Multisource VHR Images via Deep Siamese Convolutional Multiple-Layers Recurrent Neural Network. *IEEE Trans. Geosci. Remote Sens.* **2020**, *58*, 2848–2864. [CrossRef]
14. Scott, G.J.; Marcum, R.A.; Davis, C.H.; Nivin, T.W. Fusion of Deep Convolutional Neural Networks for Land Cover Classification of High-Resolution Imagery. *IEEE Geosci. Remote Sens. Lett.* **2017**, *14*, 1638–1642. [CrossRef]
15. Scott, G.J.; Hagan, K.C.; Marcum, R.A.; Hurt, J.A.; Anderson, D.T.; Davis, C.H. Enhanced Fusion of Deep Neural Networks for Classification of Benchmark High-Resolution Image Data Sets. *IEEE Geosci. Remote Sens. Lett.* **2018**, *15*, 1451–1455. [CrossRef]
16. Anderson, D.T.; Scott, G.J.; Islam, M.; Murray, B.; Marcum, R., Fuzzy Choquet Integration of Deep Convolutional Neural Networks for Remote Sensing. In *Computational Intelligence for Pattern Recognition*; Pedrycz, W.; Chen, S.M., Eds.; Springer International Publishing: Cham, Switzerland, 2018; pp. 1–28. [CrossRef]
17. Hurt, J.A.; Scott, G.J.; Davis, C.H. Comparison of Deep Learning Model Performance between Meta-Dataset Training Versus Deep Neural Ensembles. In Proceedings of the IGARSS 2019–2019 IEEE International Geoscience and Remote Sensing Symposium, Yokohama, Japan, 28 July–2 August 2019; pp. 1326–1329. [CrossRef]
18. Bajkowski, T.M.; Scott, G.J.; Hurt, J.A.; Davis, C.H. Extending Deep Convolutional Neural Networks from 3-Color to Full Multispectral Remote Sensing Imagery. In Proceedings of the 2020 IEEE International Conference on Big Data (Big Data), Virtual, 10–13 December 2020; pp. 3895–3903. [CrossRef]

19. Baghdasaryan, L.; Melikbekyan, R.; Dolmajain, A.; Hobbs, J. Deep Density Estimation Based on Multi-Spectral Remote Sensing Data for In-Field Crop Yield Forecasting. In Proceedings of the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, New Orleans, LA, USA, 18–24 June 2022; pp. 2014–2023.
20. Senecal, J.J.; Sheppard, J.W.; Shaw, J.A. Efficient Convolutional Neural Networks for Multi-Spectral Image Classification. In Proceedings of the 2019 International Joint Conference on Neural Networks (IJCNN), Budapest, Hungary, 14–19 July 2019; pp. 1–8. [CrossRef]
21. Albertini, C.; Gioia, A.; Iacobellis, V.; Manfreda, S. Detection of Surface Water and Floods with Multispectral Satellites. *Remote Sens.* **2022**, *14*, 6005. [CrossRef]
22. Chen, H.; Shi, Z. A Spatial-Temporal Attention-Based Method and a New Dataset for Remote Sensing Image Change Detection. *Remote Sens.* **2020**, *12*, 1662. [CrossRef]
23. Gargees, R.S.; Scott, G.J. Deep Feature Clustering for Remote Sensing Imagery Land Cover Analysis. *IEEE Geosci. Remote Sens. Lett.* **2020**, *17*, 1386–1390. [CrossRef]
24. Yang, A.; Hurt, J.A.; Veal, C.T.; Scott, G.J. Remote Sensing Object Localization with Deep Heterogeneous Superpixel Features. In Proceedings of the 2019 IEEE International Conference on Big Data (Big Data), Los Angeles, CA, USA, 9–12 December 2019; pp. 5453–5461. [CrossRef]
25. Rashkovetsky, D.; Mauracher, F.; Langer, M.; Schmitt, M. Wildfire Detection From Multisensor Satellite Imagery Using Deep Semantic Segmentation. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2021**, *14*, 7001–7016. [CrossRef]
26. Wang, Y.; Li, Z.; Zeng, C.; Xia, G.S.; Shen, H. An Urban Water Extraction Method Combining Deep Learning and Google Earth Engine. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2020**, *13*, 769–782. [CrossRef]
27. Nejad, S.M.M.; Abbasi-Moghadam, D.; Sharifi, A.; Farmonov, N.; Amankulova, K.; László, M. Multispectral Crop Yield Prediction Using 3D-Convolutional Neural Networks and Attention Convolutional LSTM Approaches. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2023**, *16*, 254–266. [CrossRef]
28. Le, X.H.; Ho, H.V.; Lee, G.; Jung, S. Application of Long Short-Term Memory (LSTM) Neural Network for Flood Forecasting. *Water* **2019**, *11*, 1387. [CrossRef]
29. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. In Proceedings of the International Conference on Learning Representations, Banff, AB, Canada, 14–16 April 2014.
30. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778. [CrossRef]
31. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. In Proceedings of the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 1–9.
32. Huang, G.; Liu, Z.; Weinberger, K.Q.; van der Maaten, L. Densely connected convolutional networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4700–4708.
33. Tan, M.; Le, Q.V. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. *arXiv* **2019**, arXiv.1905.11946.
34. Sundermeyer, M.; Ney, H.; Schlüter, R. From Feedforward to Recurrent LSTM Neural Networks for Language Modeling. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2015**, *23*, 517–529. [CrossRef]
35. Bahdanau, D.; Cho, K.; Bengio, Y. Neural Machine Translation by Jointly Learning to Align and Translate. *arXiv* **2014**, arXiv.1409.0473.
36. Graves, A.; Jaitly, N. Towards End-To-End Speech Recognition with Recurrent Neural Networks. In Proceedings of the 31st International Conference on Machine Learning, Bejing, China, 22–24 June 2014; PMLR Volume 32, pp. 1764–1772.
37. Greff, K.; Srivastava, R.K.; Koutnik, J.; Steunebrink, B.R.; Schmidhuber, J. LSTM: A Search Space Odyssey. *IEEE Trans. Neural Netw. Learn. Syst.* **2017**, *28*, 2222–2232. [CrossRef] [PubMed]
38. Xingjian, S.; Chen, Z.; Wang, H.; Yeung, D.Y.; Wong, W.k.; Woo, W.C. Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting. *arXiv* **2015**, arXiv:1506.04214.
39. Hu, W.S.; Li, H.C.; Pan, L.; Li, W.; Tao, R.; Du, Q. Spatial–Spectral Feature Extraction via Deep ConvLSTM Neural Networks for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2020**, *58*, 4237–4250. [CrossRef]
40. Yin, J.; Qi, C.; Chen, Q.; Qu, J. Spatial-Spectral Network for Hyperspectral Image Classification: A 3-D CNN and Bi-LSTM Framework. *Remote Sens.* **2021**, *13*, 2353. [CrossRef]
41. Zijlma, A. The Weather and Climate in Tanzania. 2020. Available online: https://www.tripsavvy.com/tanzania-weather-and-average-temperatures-4071465 (accessed on 9 August 2023).

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.