

---

# Lightweight, Pre-trained Transformers for Remote Sensing Timeseries

---

**Gabriel Tseng**<sup>\*</sup>      **Ivan Zvonkov**<sup>\*</sup>      **Mirali Purohit**  
 McGill University      University of Maryland, College Park      Arizona State University  
 Mila – Quebec AI Institute

**David Rolnick**  
 McGill University  
 Mila – Quebec AI Institute

**Hannah Kerner**  
 Arizona State University

## Abstract

Machine learning algorithms for parsing remote sensing data have a wide range of societally relevant applications, but labels used to train these algorithms can be difficult or impossible to acquire. This challenge has spurred research into self-supervised learning for remote sensing data aiming to unlock the use of machine learning in geographies or application domains where labelled datasets are small. Current self-supervised learning approaches for remote sensing data draw significant inspiration from techniques applied to natural images. However, remote sensing data has important differences from natural images – for example, the temporal dimension is critical for many tasks and data is collected from many complementary sensors. We show that designing models and self-supervised training techniques specifically for remote sensing data results in both smaller and more performant models. We introduce the **Pretrained Remote Sensing Transformer** (Presto), a transformer-based model pre-trained on remote sensing pixel-timeseries data. Presto excels at a wide variety of globally distributed remote sensing tasks and outperforms much larger models. Presto can be used for transfer learning or as a feature extractor for simple models, enabling efficient deployment at scale.

## 1 Introduction & Related Work

Increasing machine learning capabilities along with the vast amount of remote sensing data being collected provides many opportunities for societally beneficial outcomes ranging from tracking progress on sustainable development goals [1] to improved weather forecasting [2, 3] to disaster management [4]. Datasets from remote sensing often have very few labels [5], can contain high levels of label uncertainty [6, 7] and are frequently unavailable for under-resourced geographies [8, 9], leading to poor global generalization [7].

The limited availability of labeled datasets but plentiful unlabeled data has spurred the investigation of self-supervised learning algorithms tailored to remote sensing [10, 11, 12, 13, 14]. Previous approaches primarily treat remote sensing data as analogous to natural imagery, and therefore attempt to co-opt methods and architectures originally designed for natural imagery (i.e., ground-level photography) – for example, by using a ResNet [15] backbone [12, 13, 14], or by adapting masked autoencoding for image classification [16] to satellite imagery [10, 11].

However, remote sensing data differs from natural imagery in two important ways:

---

<sup>\*</sup>Equal contribution

- **Highly Informative Temporal Dimension:** The Earth’s highly dynamic nature [7] and the relatively coarse resolution of freely available satellite data means that in remote sensing, the temporal dimension is critical for many downstream tasks [17]. A common approach by remote sensing practitioners is to train single pixel-timeseries models [18, 19, 17, 20, 21, 22, 23, 24]. Current self-supervised approaches do not address the use case of these single pixel-timeseries models. Current models require spatial information as input and generally ingest at most 3 timesteps [11], with many designed only for single timestep inputs [10, 12, 14].
- **Variety of Sensors:** Many different sensors capture observations of the same place and time on Earth. Plentiful research shows that leveraging these complementary sensors can significantly improve model performance [25, 26, 27, 28] and may be used to improve self-supervised learning outcomes [29]. However, existing image-based pre-training approaches are generally trained on at most a single remote sensing data product [11] and often RGB images in particular, which represent only a subset of wavelengths captured by many satellites [10, 14, 12].

To take advantage of the unique characteristics of remote sensing data, we introduce the **Pretrained Remote Sensing Transformer** (Presto), a lightweight transformer-based model designed to ingest pixel-timeseries inputs from a variety of Earth observation sensors and data products. By tailoring the self-supervised learning process to learn from multiple sensors and the temporal dimension of the data, Presto learns powerful representations of remote sensing data. Presto excels across a wide range of remote sensing tasks, including image-based tasks where the temporal dimension is completely absent. Presto is also robust to missing input sensors, excelling in tasks where only a subset of the input channel sensors are available.

In order to be useful, models built for remote sensing data typically must be used to make contiguous geospatial predictions over millions (or billions) of samples to form a predicted map. Thus the computational performance of models is a critical consideration, especially for under-resourced institutions and research [20, 30, 31, 32, 33]. Presto was designed with practical deployment in mind and has  $\sim 1000 \times$  fewer trainable parameters than ScaleMAE and SatMAE [10, 11]. Presto can be fine-tuned on a 2017 Macbook Pro, while SatMAE requires 8 NVIDIA v100 GPUs for fine-tuning. This makes Presto accessible to practitioners without access to large amounts of compute.

In summary, the main contributions of this work are:

- We introduce Presto, a lightweight transformer-based model designed for Earth observation pixel-timeseries data together with a novel self-supervised pre-training methodology designed to leverage the structure in multi-sensor timeseries data.
- We show that Presto outperforms or matches state-of-the-art models across a wide variety of geographies, dataset sizes, and task types, despite having  $1000 \times$  fewer trainable parameters.
- We find that Presto outperforms other algorithms even on tasks that do not incorporate timeseries or multi-sensor information, showing that leveraging such information during pre-training confers a significant advantage in learned representations.

All code and data used to train and evaluate Presto is available at [github.com/nasaharvest/presto](https://github.com/nasaharvest/presto).

## 2 Method

We aim to learn a model,  $f$ , which can learn useful representations in a self-supervised manner given unlabelled remote sensing pixel-timeseries data,  $x$ . This model can then be fine-tuned for a wide variety of downstream remote sensing tasks. Importantly, while we leverage the structure of multi-sensor pixel-timeseries data for self-supervised pre-training, the model can then be fine-tuned for other input data formats with a different number of timesteps or sensors (such as Sentinel-1 radar observations or Sentinel-2 multispectral imagery) than the pre-training input format.

Our approach is based on the masked autoencoding framework [16], in which the network architecture includes both an encoder ( $f$ ) and a decoder ( $g$ ). During pre-training, a mask is applied to  $x$  (yielding a masked version of the input,  $x_m$ , and an unmasked version,  $x_u$ ). The unmasked input is passed to the encoder, and the encoder’s output is passed to the decoder. The model is trained to minimize the difference between the original masked input and the decoder’s output:

$$\operatorname{argmin}_{f,g} l(x_m, g(f(x_u))),$$

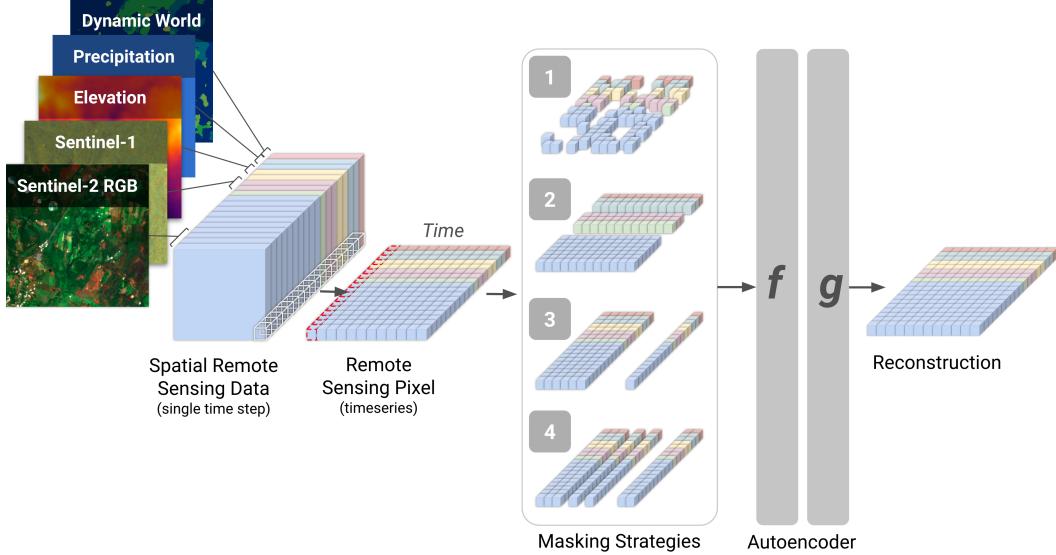


Figure 1: **Presto learns from structurally-masked remote sensing pixel-timeseries.** Specifically, we construct a multi-sensor remote sensing pixel-timeseries, and randomly select one of the four masking strategies described in Section 2.3 to mask the data. The encoder-decoder model is then trained to reconstruct the unmasked timeseries. At fine-tuning time, we discard the decoder and only use the encoder’s output. The downstream task may have incomplete inputs (missing timesteps or sensors) since the encoder is specifically trained on such inputs.

where  $l$  is a distance metric (such as the mean squared error metric used by [16]).

At fine-tuning time, we discard  $g$  and only use  $f$  (either as a feature extractor or a fine-tuneable model) for downstream tasks. In the sections below, we discuss how Presto adapts this general framework for multi-sensor remote sensing timeseries data. An overview of the Presto pre-training methodology is available in Figure 1.

## 2.1 Categorical and Continuous Timeseries Inputs

A key difference between Presto and other self-supervised approaches [11, 34, 16, 10] is that Presto is pre-trained using pixel-timeseries instead of images. From a remote sensing perspective, there are several advantages to processing pixel-timeseries instead of entire images:

- Many remote sensing applications are specifically designed for pixel-timeseries, particularly when change over time is critical [18, 19, 17, 20, 21, 22, 23, 24]. Presto allows self-supervised learning to be leveraged in this domain.
- Remote sensing labels are often collected at points or irregular polygons [35, 36, 37], and spatially complete labels can be challenging to obtain [38]. Image-based methods need to consider missing labels [38], while pixel-timeseries can ingest only the pixels where labels are present (reducing fine-tuning complexity).
- Pixel-timeseries are significantly smaller than images (a typical input for Presto is a 12 timestep  $\times$  19 channel timeseries, whereas ScaleMAE and SatMAE ingest  $224 \times 224 \times 3$  images). This allows for much smaller models to be trained (Presto has  $\sim 1000\times$  fewer trainable parameters than ScaleMAE and SatMAE), which is important for their practical deployment [20, 30, 31, 32, 33].

Vision transformers typically process an input image by dividing the image into several patches of size  $H \times W \times 3$  (or  $H \times W \times 11$  for the multi-spectral SatMAE model) [11, 34, 16, 10]. These image patches then act as input tokens into the model. Instead of focusing on the spatial dimension, Presto prioritizes learning patterns in the temporal dimension. Presto processes an input pixel-timeseries by dividing the timeseries into timestep-patches (where each timestep-patch consists of 1 timestep and a subset of total input channels). These timestep-patches then act as input tokens into the model. For each input timestep-patch, we group subsets of channels based on semantic relationships (e.g.,

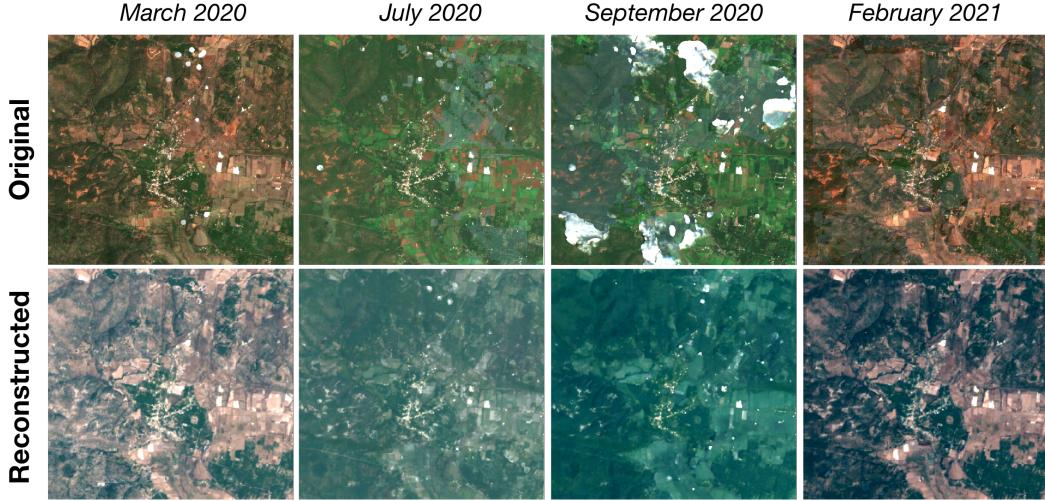


Figure 2: **Presto learns to reconstruct channels that are completely masked in a spatially cohesive manner.** In this experiment, we mask the Sentinel-2 RGB channels; Presto is able to reconstruct these channels even when they are absent from the input. For the September 2020 timestep, Presto is able to impute missing data due to cloud artefacts present in the original input data. We highlight that the reconstructions are spatially consistent even though Presto only receives single pixel inputs.

similar wavelengths detected by the same sensor, such as the Sentinel-2 shortwave infrared bands). A unique linear transformation is learned for each channel group.

In addition, Presto pre-training leverages both directly sensed remote sensing inputs (such as Sentinel-2 multispectral images) and derived inputs (such as time-evolving land cover classification maps [39] and topography maps [40]). Some derived products consist of categorical values (e.g., the land cover classification maps). In this case, the linear projection consists of a learnable embedding.

## 2.2 Learnable and Fixed Metadata Encodings

Unlike natural images, in which the data and its label are self contained, remote sensing labels are inherently associated to a place and time on Earth (i.e., a latitude/longitude and timestamp). In addition, while each patch in a natural image contains the same RGB channels, Presto’s patches represent channels from different remote sensing data products. We therefore want to communicate to the model: (i) the location of the datapoint, and a patch’s (ii) timestamp and (iii) channel group.

**Latitude & Longitude** We add a token representing the latitude and longitude of the datapoint, converted to Cartesian coordinates:

$$t_{\text{loc}} = h([\cos(\text{lat}) \times \cos(\text{lon}), \cos(\text{lat}) \times \sin(\text{lon}), \sin(\text{lat})])$$

where  $h$  is a linear transformation of the input Cartesian coordinate vector.

**Timestamp** In addition to the relative positional encoding used by transformer models [41], we add a positional encoding that represents the month being captured by each token. This is done because we expect timesteps from similar months will be similar even if they are from different years. We assign an integer to each month ranging from 0 to 11 and our month encoding is given by:

$$ME_{\text{month},2i} = \sin\left(\frac{2\pi \times \text{month}}{12}\right) \quad ME_{\text{month},2i+1} = \cos\left(\frac{2\pi \times \text{month}}{12}\right)$$

**Channel Group** Finally, each token is associated with a set of input channels. In multi-spectral SatMAE[11], a fixed encoding is used to communicate input-band information. A fixed encoding can be used because only input data from one sensor (Sentinel-2) is used, with different channels representing different wavelengths. Since Presto uses a number of different remote sensing products, we apply a learnable encoding for each channel-group.

Channel Groups	Random	Random Timesteps	Contiguous Timesteps	F1 Score
✓				0.538
	✓			0.555
		✓		0.555
			✓	0.579
✓	✓	✓	✓	<b>0.592</b>

**Table 1: Structured masking strategies yield the best downstream performance.** We measured the F1 score of Presto with linear probing ( $\text{Presto}_R$ ) on the CropHarvest multiclass validation task. A combination of structured strategies performed best (outperforming the “Random” masking commonly used by other MAE methods).

The complete encoding is a concatenation of the positional, month and learned channel encodings.

### 2.3 Pre-training via Structured Masking

A goal of this model is to perform well even with incomplete inputs (i.e., when there are fewer or missing timesteps of data, when channels are missing, or both). When constructing  $x_m$  and  $x_u$ , we therefore tailor the masking strategies to encourage the model to learn representations that perform well specifically when given a subset of bands or timesteps for downstream tasks. Specifically, for a  $T \times D$  input of  $T$  timesteps and  $D$  total input channels, we use the following masking techniques (where Presto considers a token to be a  $1 \times d$  input (a single timestep of  $d$  grouped channels)):

1. **Random:**  $(t \times d)$  masked values, with  $t < T$  and  $d < D$
2. **Channel-groups:**  $(T \times d)$  masked values, with  $d < D$
3. **Contiguous timesteps:**  $(t \times D)$  masked values, with  $t < T$
4. **Random timesteps:**  $(t \times D)$  masked values, with  $t < T$

For each instance, we randomly sample from the above strategies to construct a mask. Table 1 shows results from ablating the masking strategies. Unlike other masked-autoencoder methods [11, 16], structured masking strategies outperform random masking. We show that Presto can reconstruct entirely missing channel-groups in Figure 2.

Since the masked-autoencoder model is trained on both categorical and continuous timeseries inputs, the model’s loss must also be calculated using both categorical and continuous reconstructions. This is achieved using the following loss function, which balances the loss for every batch so that each reconstructed value receives the same weighting in the final loss:

$$\mathcal{L}_{total} = \mathcal{L}_{MSE} + \lambda \frac{N_{cat}}{N_{cont}} \mathcal{L}_{CE} \quad (1)$$

$\mathcal{L}_{MSE}$  is the mean squared error reconstruction loss used for the continuous values,  $\mathcal{L}_{CE}$  is the cross entropy loss used for the categorical values,  $N_{cont}$  is the number of masked continuous values and  $N_{cat}$  is the number of masked categorical values (in the batch).  $\lambda$  is a hyperparameter, which we set to 2 in all our experiments.

### 2.4 Pre-training Data

In this section, we describe the data used to pre-train Presto. The pre-training data is highly diverse in terms of sensor types and in terms of geographic and semantic diversity. We show in Section 4 that the full input used for pre-training is not necessary for strong performance on downstream tasks. This allows Presto to be applied to tasks even when a small subset of data timesteps and channels (relative to the pre-training data) is available.

Presto is trained on 12 months of data, where each month is represented by a monthly timestep (similar to the approach adopted in [36]). For each pixel-timeseries, we exported 2 years of data (spanning 2020 and 2021), and randomly selected a 12-month interval to pass to the model.

Derived data products (products which result from the analysis of lower level data, such as topography [42]) can significantly improve model performance [35, 24]. We therefore pre-train Presto on both directly-sensed and derived Earth observation products, exported using Google Earth Engine [43]:

- Sentinel-2 Multispectral images. We removed the 60m resolution bands, yielding 10 bands with 10m and 20m resolution with channels in the visible, near-infrared and short-wave infrared range.

Table 2: **We evaluate Presto on a wide variety of downstream tasks.** Specifically, there is diversity both in terms of input-data composition (varying numbers of timesteps and channels), geographic area and training set size.

Dataset	Task	Region	Timesteps	Channels	Training size
CropHarvest	Segmentation	Kenya			1,345
		Brazil	12	19	203
		Togo			1,319
TreeSat	Multi-Label Classification	Germany	1	2 (S1) 10 (S2)	45,337
EuroSat	Classification	Europe	1	3 (RGB) 10 (MS)	21,600
Fuel Moisture	Regression	USA	3	19	1,578
Algae Blooms	Regression	USA	12	19	777

- Sentinel-1 Synthetic Aperture Radar observations. We included the VV (emit and receive at vertical polarization) and VH (emit at vertical and receive at horizontal polarization) bands.
- ERA5 Climate Reanalysis Meteorological data. We included monthly total precipitation and temperature at 2 metres above the ground.
- Topography data, from the Shuttle Radar Topography Mission’s Digital Elevation Model. We included the elevation and slope of each pixel.
- Dynamic World Land Cover classes[39]. This product consists of land cover classes produced for every non-cloudy Sentinel-2 image. We took the mode of classes for all timesteps in a month.

Remote sensing models can be deployed in a wide range of geographies, for which few datapoints are available at fine-tuning time [8, 44]. We therefore aimed to collect a globally representative pre-training dataset, so that the final trained model could be applied in a wide range of geographies.

We achieved this by following the sampling strategy used by Dynamic World[39]. We divided the Earth into three regions: the Western Hemisphere and two regions in the Eastern Hemisphere. These regions are further divided into ecoregions, and stratified samples are gathered from each region using land cover classes as sampling strata. Each sample represents a  $510 \times 510$  pixel tile with 10 meter per pixel spatial resolution. To extract pixel-timeseries from each sample, we grid-sampled 2,500 pixels, yielding a total of 21,535,000 pixel samples.

### 3 Evaluation

We aim to demonstrate the utility of Presto for a diversity of tasks, geographic task-locations, input data modalities and downstream dataset sizes - this diversity is demonstrated in Table 2. For downstream evaluation, we take the encoder-decoder model learned during pre-training and discard the decoder. We evaluate the performance of three different models ( $\text{Presto}_R$ ,  $\text{Presto}_{RF}$ ,  $\text{Presto}_{FT}$ ) built on top of Presto’s encoder:

- **Feature extraction.** Rolf et al. [45] demonstrated the utility of neural networks as feature-extractors on top of which computationally efficient classifiers could be trained. The models  $\text{Presto}_R$  and  $\text{Presto}_{RF}$  consist respectively of regression and random forest classifiers trained on top of Presto’s embeddings. Here, only the regression/random forest is trained, not the encoder itself.
- **Fine-tuning.** The model  $\text{Presto}_{FT}$  consists of the encoder, with a linear transformation on the global-average of the encoder’s output tokens. This entire model (the encoder and the linear transformation) is fine-tuned on the evaluation tasks, using fixed hyperparameters with a subset of the training data used for validation when early stopping.

#### 3.1 Timeseries Tasks

**Crop type Segmentation** The CropHarvest [36] evaluation datasets consist of classifying pixels as (i) maize in Kenya, (ii) coffee in Brazil and (iii) crop or non-crop in Togo. We compare Presto to the baselines which accompany CropHarvest, as well as to Task-Informed Meta-Learning [46], a meta-learning method which achieves state-of-the-art results on these datasets. We additionally use

Model	Kenya	Brazil	Togo	Mean
Random Forest	0.559	0.000	0.756	0.441
MOSAIKS-1D	0.821	0.676	0.689	0.729
TIML[46]	0.838	0.835	0.732	0.802
Presto <sub>R</sub> no DW	0.839 <b>0.852</b>	0.939 <b>0.959</b>	<b>0.810</b> 0.770	<b>0.863</b> 0.860

Table 3: F1 scores on the CropHarvest tasks. Presto outperforms TIML despite being pre-trained in a fully self-supervised manner (TIML is pre-trained on CropHarvest[36]). TIML and MOSAIKS-1D do not receive Dynamic World as input, so we evaluated Presto without it for fair comparison.

Model	Data	Weighted		Micro	
		$F_1$	mAP	$F_1$	mAP
MLP [48]		<b>51.97</b>	64.19	<b>54.59</b>	65.83
LightGBM [48]	S2	48.17	61.99	52.52	61.66
Presto <sub>RF</sub>		46.61	<b>66.93</b>	50.62	<b>67.72</b>
MLP [48]		10.09	29.42	12.82	33.09
LightGBM [48]	S1	11.86	32.79	14.07	35.11
Presto <sub>RF</sub>		<b>27.55</b>	<b>49.86</b>	<b>33.03</b>	<b>51.42</b>

Table 4: Classification results on the TreeSatAI dataset. We compare Presto to the benchmark models released alongside TreeSatAI. The MLP consists of 3 layers, and has 1.13M-1.31M parameters (compared to 271k parameters for the Presto encoder)

the FAO indicative crop classification labels as a multiclass validation task. No test data from the evaluation datasets is present in the validation task.

**Fuel Moisture** The Western USA live fuel moisture dataset [35] consists of estimating live fuel moisture content in the Western United States (a regression task). We compare our method to the physics-assisted neural network which accompanied the dataset. This baseline used 5-fold validation to evaluate model performance – we use a geographically separated test set to evaluate the model.

**Algae Blooms** The harmful algae blooms dataset [47] consists of estimating the severity of cyanobacterial algal blooms in the United States. We use a subset of the data (in the Midwestern U.S.). Since the dataset was originally released as part of a competition, the test data is not available. In addition, competitors could download a range of earth observation datasets to train their models, making direct comparisons difficult. We benchmark against a logistic regression and a random forest (since the winning solution used a tree-based method), and use a geographically separated test set.

### 3.2 Image-based Tasks

A common paradigm in remote sensing is to consider satellite as static-in-time image data. We therefore evaluate Presto against two image-based datasets. Since Presto is designed to ingest pixel-timeseries (and not single timestep images, as is the case here), for the datasets below we sample 9 pixels from each image and pass each of the single pixels to the model; the reported results are the average of the predictions for the 9 pixels.

**TreeSatAI** The TreeSatAI dataset consists of classifying tree species (out of 20 possible species) from forestry images in Germany [48]. We use the train and test splits provided by [48], and compare Presto to the deep-learning and tree-based baselines provided alongside the dataset. The benchmark models [48] measure the effectiveness of S2-only or S1-only models, so we do the same. Since this is a multilabel dataset, we take the mean of the 9 pixel-predictions to find the predicted classes.

**EuroSAT** The EuroSAT dataset consists of classifying images in Europe as belonging to one of 10 landcover classes [5]. Exact dataset splits are not provided, so we split the data according to ratios provided by SatMAE [11]. We compare Presto to SatMAE and ScaleMAE[10] by using the KNN-classifier approach used by ScaleMAE. We take the mode of the 9 pixel-predictions.

### 3.3 Baselines

In addition to the task-specific baselines described above, we benchmark Presto against:

Table 5: Accuracy results on the EuroSAT dataset for Presto, and for the SatMAE and ScaleMAE models, as reported in [10] (using the results for reduced resolution inputs). As in [10], we use a K-Nearest-Neighbours classifier with  $k$  neighbours based on the learned representations of the respective models. Both SatMAE and ScaleMAE receive  $28\times$  more input pixels than Presto and are specifically designed to ingest remote-sensing images. We evaluate Presto against both the RGB and multi-spectral versions of the EuroSat dataset – it achieves the best results (by a significant margin).

Model	# input pixels	$k = 5$	$k = 20$	$k = 100$
SatMAE@16	256	0.729	0.727	0.695
ScaleMAE@16	256	0.723	0.721	0.676
Presto (RGB)	9	0.824	0.804	0.752
Presto (MS)	9	<b>0.864</b>	<b>0.841</b>	<b>0.795</b>

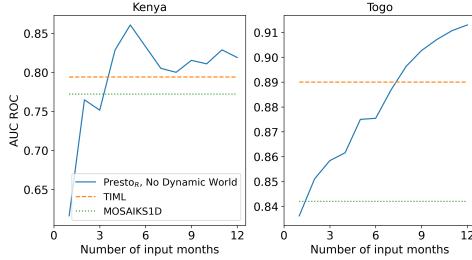


Figure 3: **Presto is robust to incomplete inputs.** We measure the AUC ROC score of Presto with Linear probing ( $\text{Presto}_R$ ) on the CropHarvest dataset when no Dynamic World input is passed, and with a subset of input months (the x-axis).  $\text{Presto}_R$  recovers the performance of MOSAIKS-1D and TIML with 12 months of input given only a subset of input months.

- **Random Forests:** Random Forests are powerful baselines in remote sensing; they remain competitive with state-of-the-art methods [21, 8, 49] and tree-based methods – especially random forests – are commonly deployed in large-scale machine learning for remote sensing [50, 24, 30, 51].
- **MOSAIKS-1D:** We adapt MOSAIKS [45] for timeseries data. MOSAIKS-1D uses patches from the pre-training dataset and convolves over the temporal dimension instead of the spatial dimension. We benchmark MOSAIKS-1D on all timeseries evaluation tasks. Because this approach does not work for categorical inputs, we ignore the Dynamic World input when using MOSAIKS-1D.
- **Fully Supervised Presto:** To disentangle the effects of the model architecture from the pre-training regimen, we fine-tune a Presto architecture starting from randomly initialized weights.

## 4 Results

Overall, Presto excels at a variety of tasks (classification tasks and regression tasks) in a variety of geographies (spanning 4 continents and 38 countries). Presto is performant whether the whole model is fine-tuned (Table 6) or used as a feature extractor for simple models (Tables 3, 4 and 5). We highlight that for many of the datasets, we benchmark Presto against the state of the art model for that task – even as a feature extractor, Presto outperforms the state of the art in many cases (Tables 3, 4 and 6). In addition, whereas previous self-supervised learning approaches primarily consider downstream land-cover mapping tasks [11, 10], we demonstrate Presto’s performance across a wide range of tasks ranging from tree species classification to algae bloom estimation.

Fine-tuning Presto is computationally efficient; while comparable methods require a cluster of GPUs at fine-tuning time [11], we can fine-tune Presto on a single GPU or CPU. For example, Presto can be fine-tuned on the Fuel Moisture evaluation task on a 2017 MacBook Pro in less than 6 minutes, making Presto accessible to practitioners without access to significant computational resources.

**Presto is performant even with missing data** Presto is pre-trained on a variety of diverse remote sensing products (see Section 2.4). However, these data products may not be available when fine-tuning for a downstream task; for example, only a single timestep may be available [5, 48]. We therefore evaluate Presto when it receives:

- **A subset of timesteps:** We evaluate Presto when it receives only a subset of timesteps compared to the 12 timesteps used for pre-training. Specifically, Presto receives 3 input timesteps for the fuel moisture task (Table 6) and only a single input timestep for the EuroSat and TreeSatAI tasks

Table 6: RMSE results on the regression tasks. The algae bloom literature baseline is not directly comparable as we only use a subset of the training data and test data is not made available – we include it as an illustrative baseline. Best results are **highlighted blue**, with second best results in **bold**. Models have a high variance in performance across tasks – we therefore calculate the mean difference in RMSE from the linear regression baseline across both tasks. Presto performs most consistently, both when used as a feature-extractor for random forests and when fine-tuned.

	Fuel Moisture	Algae Blooms	Mean difference
Literature baseline	25	0.761*	
Linear Regression	28.20	<b>0.850</b>	0%
Random Forest	<b>23.63</b>	1.210	13.1%
MOSAIKS-1D	29.71	2.281	86.85%
Fully Supervised Presto	26.75	1.063	9.96%
Presto <sub>RF</sub>	<b>24.72</b>	0.887	-4.00%
Presto <sub>FT</sub>	25.35	<b>0.816</b>	-7.05%

(Tables 4 and 5). In all cases, Presto remains performant. We also evaluate Presto when a subset of input months are passed for the CropHarvest dataset (Figure 3) – Presto rapidly recovers the performance of TIML and MOSAIKS-1D (with all input months).

- **A subset of input bands:** We also evaluate Presto when it only receives a subset of the training input bands. For the EuroSat task (Table 5), Presto receives either the full Sentinel-2 input, or only the RGB bands (which represents only a single token, since only one timestep is available for this task). Similarly, we evaluate Presto when it receives either Sentinel-2 or Sentinel-1 data for the TreeSatAI task (Table 4) In both cases, Presto outperforms methods designed to ingest single timestep, single sensor data. Finally, Presto remains performant (compared to a full input) when passed a subset of channels, seeing a small performance dip compared to a complete input on the CropHarvest dataset (Table 3).

**Presto outperforms other MAE methods designed for satellite data** We compare Presto to SatMAE and ScaleMAE by emulating the linear-probing technique used by ScaleMAE [10] (using a K-nearest-neighbours classifier on the output of the encoder) on the EuroSat dataset [5]. We find that Presto outperforms SatMAE and ScaleMAE even when receiving significantly less input data (Table 5). We highlight that EuroSat is a single timestep imagery dataset (and is therefore in the modality that SatMAE and ScaleMAE were designed to ingest), whereas Presto is designed for pixel-timeseries. In addition, we re-emphasize the difference in model size of Presto compared to ScaleMAE and SatMAE: Presto consists of 556k trainable parameters, while ScaleMAE has 332M trainable parameters and SatMAE has 329M trainable parameters.

We hypothesize that this performance stems from Presto’s ability to reconstruct missing channels and timesteps (Figure 2). This allows it to infer information in these dimensions, improving downstream performance for sparse inputs.

**Presto’s performance stems from both the architecture and the pre-training regime** We compare Presto with self-supervision to a fully supervised Presto model (a randomly initialized Presto architecture fine-tuned on the downstream tasks) in Table 6. We find that self-supervised Presto outperforms the fully-supervised model both when it is fine-tuned and as a feature-extractor for a random forest, illustrating the importance of the pre-training for downstream performance.

## 5 Conclusion

In conclusion, we present Presto, a lightweight, pre-trained timeseries focussed transformer for remote sensing. By leveraging structure unique to remote sensing data (specifically, (i) an important temporal dimension, (ii) associated metadata and (iii) a diversity of sensors), we are able to train an extremely lightweight model which achieves state-of-the-art results in a wide variety of globally distributed evaluation tasks. Computational efficiency is of paramount importance in remote sensing

settings; we demonstrate that strong performance can be achieved while respecting this constraint, and that self-supervised learning approaches can provide significant benefits even for small models.

### Acknowledgements

This work was supported by NASA under the NASA Harvest Consortium on Food Security and Agriculture (Award #80NSSC18M0039). This research was enabled in part by compute resources provided by Mila (mila.quebec).

## References

- [1] Bruno Ferreira, Muriel Iten, and Rui G Silva. Monitoring sustainable development by means of earth observation data and machine learning: A review. *Environmental Sciences Europe*, 2020.
- [2] Stephen English, Tony McNally, Niels Bormann, Kirsti Salonen, Marco Matricardi, Andras Moranyi, Michael Rennie, Marta Janisková, Sabatino Di Michele, Alan Geer, et al. Impact of satellite data, 2013.
- [3] Paul Voosen. Europe builds ‘digital twin’ of earth to hone climate forecasts, 2020.
- [4] Pratistha Kansakar and Faisal Hossain. A review of applications of satellite earth observation data for global societal benefit and stewardship of planet earth. *Space Policy*, 2016.
- [5] Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 2019.
- [6] Patrik Olá Bressan, José Marcato Junior, José Augusto Correa Martins, Maximilian Jaderson de Melo, Diogo Nunes Gonçalves, Daniel Matte Freitas, Ana Paula Marques Ramos, Michelle Taís Garcia Furuya, Lucas Prado Osco, Jonathan de Andrade Silva, et al. Semantic segmentation with labeling uncertainty and class imbalance applied to vegetation mapping. *International Journal of Applied Earth Observation and Geoinformation*, 2022.
- [7] Ban Yifang, Peng Gong, and Chandra Gini. Global land cover mapping using earth observation satellite data: Recent progresses and challenges. *ISPRS journal of photogrammetry and remote sensing*, 2015.
- [8] Hannah Kerner, Gabriel Tseng, Inbal Becker-Reshef, Catherine Nakalembe, Brian Barker, Blake Munshell, Madhava Paliyam, and Mehdi Hosseini. Rapid response crop maps in data sparse regions. In *ACM SIGKDD Conference on Data Mining and Knowledge Discovery Workshops*, 2020.
- [9] Catherine Nakalembe, Christina Justice, Hannah Kerner, Christopher Justice, and Inbal Becker-Reshef. Sowing seeds of food security in africa. *Eos (Washington. DC)*, 102, 2021.
- [10] Colorado J Reed, Ritwik Gupta, Shufan Li, Sarah Brockman, Christopher Funk, Brian Clipp, Salvatore Candido, Matt Uyttendaele, and Trevor Darrell. Scale-mae: A scale-aware masked autoencoder for multiscale geospatial representation learning. *arXiv preprint arXiv:2212.14532*, 2022.
- [11] Yezhen Cong, Samar Khanna, Chenlin Meng, Patrick Liu, Erik Rozi, Yutong He, Marshall Burke, David B. Lobell, and Stefano Ermon. SatMAE: Pre-training transformers for temporal and multi-spectral satellite imagery. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *NeurIPS*, 2022. URL <https://openreview.net/forum?id=WBhqzpF6KYH>.
- [12] Neal Jean, Sherrie Wang, Anshul Samar, George Azzari, David Lobell, and Stefano Ermon. Tile2vec: Unsupervised representation learning for spatially distributed data. In *AAAI*, 2019.
- [13] Oscar Manas, Alexandre Lacoste, Xavier Giró-i Nieto, David Vazquez, and Pau Rodriguez. Seasonal contrast: Unsupervised pre-training from uncurated remote sensing data. In *CVPR*, 2021.
- [14] Kumar Ayush, Burak Uzkent, Chenlin Meng, Kumar Tanmay, Marshall Burke, David Lobell, and Stefano Ermon. Geography-aware self-supervised learning. In *CVPR*, 2021.
- [15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.
- [16] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *CVPR*, 2022.

- [17] Marc Rußwurm, Nicolas County, Rémi Emonet, Sébastien Lefèvre, Devis Tuia, and Romain Tavenard. End-to-end learned early classification of time series for in-season crop type mapping. *ISPRS Journal of Photogrammetry and Remote Sensing*, 2023. URL <https://www.sciencedirect.com/science/article/pii/S092427162200332X>.
- [18] Marc Rußwurm and Marco Körner. Self-attention for raw optical satellite time series classification. *ISPRS journal of photogrammetry and remote sensing*, 2020.
- [19] Gabriel Tseng, Hannah Kerner, Catherine Nakalembe, and Inbal Becker-Reshef. Learning to predict crop type from heterogeneous sparse labels using meta-learning. In *EarthVision Workshop at CVPR*, 2021.
- [20] Vivien Sainte Fare Garnot, Loic Landrieu, Sébastien Giordano, and Nesrine Chehata. Satellite image time series classification with pixel-set encoders and temporal self-attention. *CVPR*, 2020.
- [21] Charlotte Pelletier, Geoffrey I Webb, and François Petitjean. Temporal convolutional neural network for the classification of satellite image time series. *Remote Sensing*, 2019.
- [22] Sherrie Wang, Stefania Di Tommaso, Jillian M Deines, and David B Lobell. Mapping twenty years of corn and soybean across the us midwest using the landsat archive. *Scientific Data*, 2020.
- [23] Sherrie Wang, George Azzari, and David B Lobell. Crop type mapping without field-level labels: Random forest transfer and unsupervised clustering techniques. *Remote sensing of environment*, 2019.
- [24] Tomislav Hengl, Jorge Mendes de Jesus, Gerard BM Heuvelink, Maria Ruiperez Gonzalez, Milan Kilibarda, Aleksandar Blagotić, Wei Shangguan, Marvin N Wright, Xiaoyuan Geng, Bernhard Bauer-Marschallinger, et al. Soilgrids250m: Global gridded soil information based on machine learning. *PLoS one*, 2017.
- [25] Max J Steinhagen, Paul D Wagner, Balaji Narasimhan, and Björn Waske. Combining sentinel-1 and sentinel-2 data for improved land use and land cover mapping of monsoon regions. *International journal of applied earth observation and geoinformation*, 2018.
- [26] Kristof Van Tricht, Anne Gobin, Sven Gilliams, and Isabelle Piccard. Synergistic use of radar sentinel-1 and optical sentinel-2 imagery for crop mapping: A case study for belgium. *Remote Sensing*, 2018.
- [27] Andrew Whyte, Konstantinos P Ferentinos, and George P Petropoulos. A new synergistic approach for monitoring wetlands using sentinels-1 and 2 data with object-based machine learning algorithms. *Environmental Modelling & Software*, 2018.
- [28] Julien Denize, Laurence Hubert-Moy, Julie Betbeder, Samuel Corgne, Jacques Baudry, and Eric Pottier. Evaluation of using sentinel-1 and -2 time-series to identify winter land use in agricultural landscapes. *Remote Sensing*, 2018.
- [29] M Schmitt, LH Hughes, C Qiu, and XX Zhu. Sen12ms—a curated dataset of georeferenced multi-spectral sentinel-1/2 imagery for deep learning and data fusion. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 2019.
- [30] Kristof Van Tricht. Mapping crops at a global scale! what works and what doesn't? <https://blog.vito.be/remotesensing/worldcereal-benchmarking>, 2021.
- [31] Meghana Kshirsagar, Caleb Robinson, Siyu Yang, Shahrzad Gholami, Ivan Klyuzhin, Sumit Mukherjee, Md Nasir, Anthony Ortiz, Felipe Oviedo, Darren Tanner, et al. Becoming good at ai for good. In *AAAI/ACM Conference on AI, Ethics, and Society*, 2021.
- [32] Maria De-Arteaga, William Herlands, Daniel B Neill, and Artur Dubrawski. Machine learning for the developing world. *ACM Transactions on Management Information Systems (TMIS)*, 2018.
- [33] Catherine Nakalembe and Hannah Kerner. Considerations for ai-eo for agriculture in sub-saharan africa. *Environmental Research Letters*, 2023.
- [34] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021. URL <https://openreview.net/forum?id=YicbFdNTTy>.
- [35] Krishna Rao, A Park Williams, Jacqueline Fortin Flefil, and Alexandra G Konings. Sar-enhanced mapping of live fuel moisture content. *Remote Sensing of Environment*, 2020.
- [36] Gabriel Tseng, Ivan Zvonkov, Catherine Lilian Nakalembe, and Hannah Kerner. Cropharvest: A global dataset for crop-type classification. In *NeurIPS, Datasets and Benchmarks Track*, 2021. URL <https://openreview.net/forum?id=JtjzUXPEaCu>.

- [37] Niels H Batjes, Eloi Ribeiro, Ad Van Oostrum, Johan Leenaars, Tom Hengl, and Jorge Mendes de Jesus. Wosis: providing standardised soil profile data for the world. *Earth System Science Data*, 2017.
- [38] Sherrie Wang, William Chen, Sang Michael Xie, George Azzari, and David B Lobell. Weakly supervised deep learning for segmentation of remote sensing imagery. *Remote Sensing*, 2020.
- [39] Christopher F. Brown, Steven P. Brumby, Brookie Guzder-Williams, Tanya Birch, Samantha Brooks Hyde, Joseph Mazzariello, Wanda Czerwinski, Valerie J. Pasquarella, Robert Haertel, Simon Ilyushchenko, Kurt Schwehr, Mikaela Weisse, Fred Stolle, Craig Hanson, Oliver Guinan, Rebecca Moore, and Alexander M. Tait. Dynamic world, near real-time global 10 m land use land cover mapping. *Scientific Data*, Jun 2022.
- [40] SRTM 90m Digital Elevation Data. The CGIAR consortium for spatial information, 2003.
- [41] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *NeurIPS*, 2017.
- [42] CL Parkinson, A Ward, and MD King. Earth science reference handbook. *National Aeronautics and Space Administration: Washington, DC, USA*, 2006.
- [43] Noel Gorelick, Matt Hancher, Mike Dixon, Simon Ilyushchenko, David Thau, and Rebecca Moore. Google earth engine: Planetary-scale geospatial analysis for everyone. *Remote sensing of Environment*, 2017.
- [44] Vanessa Böhm, Wei Ji Leong, Ragini Bal Mahesh, Ioannis Prapas, Edoardo Nemni, Freddie Kalaitzis, Siddha Ganju, and Raul Ramos-Pollan. Sar-based landslide classification pretraining leads to better segmentation. In *Artificial Intelligence for Humanitarian Assistance and Disaster Response Workshop at NeurIPS*, 2022.
- [45] Esther Rolf, Jonathan Proctor, Tamara Carleton, Ian Bolliger, Vaishaal Shankar, Miyabi Ishihara, Benjamin Recht, and Solomon Hsiang. A generalizable and accessible approach to machine learning with global satellite imagery. *Nature communications*, 2021.
- [46] Gabriel Tseng, Hannah Kerner, and David Rolnick. TIML: Task-informed meta-learning for crop type mapping. In *AI for Agriculture and Food Systems at AAAI*, 2021.
- [47] Tick tick bloom: Harmful algal bloom detection challenge. <https://www.drivendata.org/competitions/143/tick-tick-bloom/page/649/>, 2023. Accessed: 2023-03-10.
- [48] Steve Ahlsweide, Christian Schulz, Christiano Gava, Patrick Helber, Benjamin Bischke, Michael Förster, Florencia Arias, Jörn Hees, Begüm Demir, and Birgit Kleinschmit. Treesatai benchmark archive: A multi-sensor, multi-label dataset for tree species classification in remote sensing. *Earth System Science Data*, 2023.
- [49] Mariana Belgiu and Lucian Drăguț. Random forest in remote sensing: A review of applications and future directions. *ISPRS journal of photogrammetry and remote sensing*, 2016.
- [50] Matthew C Hansen, Peter V Potapov, Rebecca Moore, Matt Hancher, Svetlana A Turubanova, Alexandra Tyukavina, David Thau, Stephen V Stehman, Scott J Goetz, Thomas R Loveland, et al. High-resolution global maps of 21st-century forest cover change. *Science*, 2013.
- [51] Stefania Di Tommaso, Sherrie Wang, Vivek Vajipey, Noel Gorelick, Rob Strey, and David B Lobell. Annual field-scale maps of tall and short crops at the global scale using gedi and sentinel-2. *arXiv preprint arXiv:2212.09681*, 2022.
- [52] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019.
- [53] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *JMLR*, 2011.

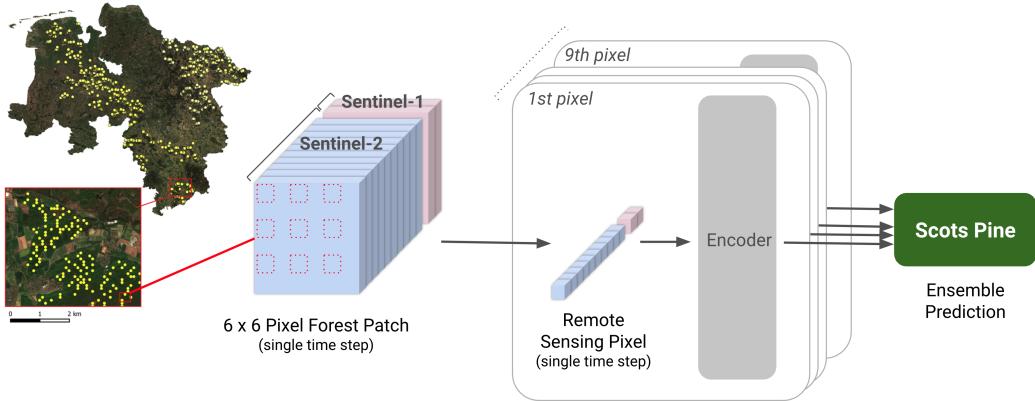


Figure 4: **Presto can be applied to image-based datasets.** We achieve this by ensembling Presto predictions for 9 grid-sampled pixels in an image, and achieve competitive results with image-based models receiving  $28\times$  as much input data.

## A Appendix

### A.1 Training Details

We base many training hyperparameter-selection decisions on what was done by the original MAE-ViT model [16]. Specifically, this consists of:

- Training the model with an AdamW optimizer [52], with a weight decay of 0.05. As per the MAE model,  $\beta$ s of (0.9, 0.95) are used.
- Applying a half-cosine schedule with a warmup to the learning rate. The maximum learning rate (0.0025) is reached after a 4 epoch warmup, and total training consist of 20 epochs. Learning rate is reduced to a minimum of 0.

We use a batch size of 8192. Self-supervised training takes 73 hours on a single NVIDIA RTX8000 GPU. However, validation takes a significant amount of time (particularly as we implement linear probing on the CropHarvest FAO validation task for each validation step). We expect that training time could be significantly reduced with the removal of the validation task.

A masking ratio of 0.5 is used; this is to ensure all the masking combinations (i.e. a full set of channel-groups or timestamps) can be combined in a single batch.

#### A.1.1 Evaluation tasks

We show the application of Presto to image-based datasets in Figure 4.

**Presto as a feature extractor** When used as a feature extractor, a Random Forest or a Regression is trained on Presto’s output embeddings. In this case, we use scikit-learn models with the default hyperparameters [53] (with the sole exception of setting `class_weight` equal to `balanced` for all classification tasks).

**Fine-tuning Presto** When fine-tuning Presto, we use the same hyperparameters across all tasks: an AdamW optimizer with a learning rate of  $3e-4$ , and a batch size of 64. We use a geographically separated validation set with early stopping, with a patience of 10.

### A.2 Complete results

In this section, we include complete results on the evaluation tasks.

- **CropHarvest Classification:** As per [36], we include both the AUC ROC and F1 score in Table 7.

Table 7: Results on the CropHarvest dataset’s evaluation tasks for Presto (with a linear regression trained on the embeddings), a Random Forest and TIML [46]. The best results are in **bold** with second best **highlighted blue**.

	Model	Kenya	Brazil	Togo	Mean
AUC ROC	Random Forest	0.578	0.941	0.892	0.803
	MOSAIKS-1D	0.772	0.839	0.842	0.818
	TIML[46]	0.794	0.988	0.890	0.890
F1 score	Presto <sub>R</sub>	<b>0.845</b>	<b>0.998</b>	<b>0.925</b>	<b>0.923</b>
	no Dynamic World	<b>0.819</b>	<b>0.995</b>	<b>0.913</b>	<b>0.909</b>
F1 score	Random Forest	0.559	0.000	0.756	0.441
	MOSAIKS-1D	0.821	0.676	0.689	0.729
	TIML[46]	0.838	0.835	0.732	0.802
F1 score	Presto <sub>R</sub>	<b>0.839</b>	<b>0.939</b>	<b>0.810</b>	<b>0.863</b>
	no Dynamic World	<b>0.852</b>	<b>0.959</b>	<b>0.770</b>	<b>0.860</b>

Table 8: Classification results on the TreeSatAI dataset. We compare Presto to the benchmark models released alongside TreeSatAI. The MLP and LightGBM models receive 36 pixels as input, compared to 9 pixels for Presto.

Model	Data	Precision	Weighted			Micro		
			Recall	<i>F</i> <sub>1</sub>	mAP	Precision	Recall	<i>F</i> <sub>1</sub>
MLP [48]		74.59	<b>42.23</b>	<b>51.97</b>	64.19	77.18	<b>42.23</b>	<b>54.59</b>
LightGBM [48]	S2	74.27	40.04	48.17	61.99	76.27	40.04	52.52
Presto <sub>RF</sub>		<b>85.95</b>	36.17	46.61	<b>66.93</b>	<b>84.27</b>	36.17	50.62
MLP [48]		33.29	7.13	10.09	29.42	63.01	7.13	12.82
LightGBM [48]	S1	37.96	8.06	11.86	32.79	55.49	8.06	14.07
Presto <sub>RF</sub>		<b>78.97</b>	<b>21.50</b>	<b>27.55</b>	<b>49.86</b>	<b>71.21</b>	<b>21.50</b>	<b>33.03</b>
								<b>51.42</b>

- **TreeSatAI Classification:** As per [48], we include all metrics (Precision, Recall, F1 Score and mAP) in Table 8. For this multi-label classification task, we follow the benchmark models in removing any tree-class covering a surface less than 7% of the tile from the labels.