Mind the Gap: Generating Imputations for Satellite Data Collections at Myriad Spatiotemporal Scopes

Paahuni Khandelwal, Daniel Rammer, Shrideep Pallickara, Sangmi Lee Pallickara
Department of Computer Science, Colorado State University, Fort Collins, Colorado, 80521
{Paahuni.Khandelwal, Daniel.Rammer, Shrideep.Pallickara, Sangmi.Pallickara}@colostate.edu

***Abstract*—Hyperspectral satellite data collections have been successfully leveraged in many domains such as meteorology, agriculture, forestry, and disaster management. There is also a collection of publicly available satellite observation networks. However, gaps in scanning frequencies and inadequate spatial resolutions limit the capabilities of geoscience applications. In this study, we target the temporal sparsity of high-resolution satellite images. In particular, we propose a novel methodology to estimate high-resolution images between scheduled scans. Our model SATnet, falls broadly within the class of Generative Adversarial Networks. SATnet allows us to generate accurate high-resolution, high-frequency satellite data at diverse spatial extents. SATnet achieves this by learning relations between a sequence of high-resolution/low-frequency satellite imageries (from Sentinel-2) and an ancillary satellite image that is high-frequency/low-resolution (from MODIS). Our benchmarks demonstrate that SATnet outperforms existing approaches such as ConvLSTMs, Dynamic Network Filter, and TrajGRU with a PSNR accuracy of 31.6. We trained and deployed SATnet over a distributed storage cluster to support the high-throughput generation of imputed satellite imagery via query evaluations. Our methodology preserves geospatial proximity and facilitates the dynamic construction of satellite imagery at a particular timestamp for arbitrary spatial scopes.**

***Index Terms*—Spatial Data, Time series analysis, deep learning, high-resolution imaging**

I. Introduction

Hyperspectral observations from satellites provide critical information to discover scientific insights and make decisions. Periodic high-resolution monitoring of the planet has been widely used in the geosciences such as meteorology, oceanography, agriculture, forestry, and cartography. Hyperspectral imaging also informs decision making processes such as disaster mitigation and land-use/land-change.

There is a rich set of publicly available satellite imagery with global coverage. NASA and the U.S. Geological Survey's Landsat program [1] provides measurements of the Earth's terrestrial and polar regions in the visible, near-infrared, short wave infrared, and thermal infrared frequencies. A scene captured by Landsat-8 covers 185km-cross-track-by-180km-along-track; the satellite system acquires up to 725 scenes every day. The European Space Agency's Sentinel-2 provides 10-meter resolution, multispectral images every 5 days [2]. NASA's MODIS satellites acquire imagery of the entire Earth's surface every 1 or 2 days [3].

Although there is a wealth of public and commercial services available, geoscientists often encounter challenges such as gaps in scanning frequencies and insufficient spatial resolutions. There has been active research in leveraging machine learning algorithms to predict missing images [4]–[9]. These approaches develop statistical models based upon surface reflectance and landcover information, or impute the missing values based upon hyper-spectral band information. However, such methods often lack scalability or encounter tractability issues when applied to larger spatial extends or cannot leverage the large amount of historical records effectively.

The crux of this study targets data discovery capabilities that bridge the gap between the actual frequencies at which remote sensing observations are made and the application-specific data retrieval requirements. In particular, our methodology targets the on-demand generation of synthetic satellite imagery for temporal gaps that arise between scans for publicly available satellite imageries. We have designed and developed a Generative Adversarial Network, SATnet, to capture and model relationships between a sequence of imageries and ancillary datasets. Our approximate query leverages models to generate imagery for gaps in satellite scanning frequencies.

A. Research Questions

In this paper, we explore the following research questions.

RQ-1: *How can we produce imputations of satellite imagery at arbitrary spatial extents and chronological timestamps?* A key consideration here is to ensure accuracy, scale, responsiveness, and throughput.

RQ-2: *How can we preserve the structural characteristics of a geolocation while accounting for temporal variations?* Preservation of structural variations that occur over time is critical for ensuring accuracy at scale.

RQ-3: *How can we reconcile, and leverage, differences in the scanning frequencies and resolutions at which remote sensing is performed?*

B. Approach Summary

This study encompasses multiple aspects to provide estimates of satellite imageries for gaps in scanning frequencies at scale. We propose SATnet, a variation of the Generative Adversarial Networks (GANs). SATnet generates imputations of satellite imageries by capturing relationships between low-frequency, high-resolution imagery, and the corresponding high-frequency, low-resolution satellite imageries. Training and execution of SATnet is integrated within a distributed storage system.

We exemplify our approach using satellite data collections from ESA's Sentinel-2 that provides a 10-meter spatial

resolution and scanning frequency of 5 days. We also use NASA's MODIS dataset acquired daily at a much lower spatial resolution (250 meters) compared to Sentinel-2. We estimate RGB bands for the days between the scheduled scans at subsecond latencies.

The aforementioned satellite systems observe the planet periodically at different spatial resolutions. These scans result in a voluminous dataset with multiple scans at different resolutions and frequencies for given geolocation. Satellite data collection is distinct from the widely researched large image corpus in computer vision such as handwritten digits, portraits, or scenes. The same location is often observed by multiple satellites, although there are variations in data acquisitions based on spatial resolutions, temporal frequencies, or types of bands (wavelengths at which surface reflectance are sensed). Finally, landscape changes and seasonality trends are unique per region yet some regions share similarities.

Our model, SATnet is tailored to the unique characteristics of satellite imagery. SATnet generates realistic estimations using GANs to leverage overall statistics by pairing a generator and discriminator. In contrast to conventional GANs [10], [11], SATnet accounts for recent observations of the target area to propagate slow-evolving attributes such as structural information (e.g., road systems and shape of buildings). To accomplish this capability, SATnet employs time-series prediction models such as Bi-CLSTM [12] as the generator.

Even over a window of a few days, variations in the RGB band values are quite normal due to weather changes or the lifecycle of plants. To cope with short-term changes, SATnet employs ancillary datasets such as MODIS satellite imageries. Although MODIS provides significantly lower resolution scans, its higher frequency scanning provides highly relevant indicators of daily change for a particular spatial extent. SATnet captures non-linear relationships between MODIS and Sentinel imageries.

We train and maintain a model whose geospatial coverage is aligned with the data dispersion schemes of the underlying data storage system. We facilitate faster model training and execution by minimizing data movements within the cluster. In our distributed storage cluster, each node maintains an instance of SATnet. To minimize data movement during retrieval for model inputs, imageries acquired from the multiple satellites are dispersed using a universal spatial dispersion scheme.

We contrast the performance of our methodology with dominant approaches for predicting spatiotemporal variations such as ConvLSTMs, Dynamic Network Filters, and TrajGRU. SATnet outperforms the aforementioned approaches with accuracy reaching a PSNR of 31.6 with a subsecond latency to retrieve the model inferred data.

C. Paper Contributions

We have designed a scalable approach for generating imputations of satellite imageries using machine learning algorithms and ancillary datasets. Our contributions include:

- Effective imputations of satellite imagery for a particular timestamp. We have designed a variation of GAN that takes observations within the desired timestamp's temporal neighborhood to generate synthetic imagery.
- SATnet is specifically designed to harness the unique characteristics of satellite imagery, which typically have a mixture of fast and slow evolving components. By leveraging topological characteristics of satellite imageries, SATnet provides improved model accuracy.
- Our methodology effectively integrates ancillary satellite low-resolution, high-frequency imagery to accommodate daily changes. Fast evolving components such as the pixel's band values are estimated by incorporating lower resolution satellite imagery with higher scanning frequencies.
- We facilitate scalable model training and execution for high-throughput query evaluation. Our methodology targets large-scale high-throughput data discovery services and reduces network communication across storage nodes during query evaluations. To ensure scalability, our system trains and maintains models over data partitions that are grouped based on their geospatial proximity.

D. Paper Organization

Section 2 describes related work. The modeling aspects of our methodology are described in Section 3. This section also describes systems aspects of our methodology that ensure scaling and distributed, decentralized query evaluation. Section 4 describes our empirical benchmarks alongside a discussion of results. Finally, our conclusions and future work are described in Section 5.

II. Related Work

Satellite imagery often contains millions of pixels per image. Capturing accurate inferences at such a high resolution is difficult. Therefore, analyses rely on an effective spatial partitioning scheme [13]. These techniques have been applied in myriad domains; effectively distributing analyses of CT scan imagery [14], extracting farmland [15], and mapping irrigation areas [16] among others. Geographic Object-Based Image Analysis (GEOBIA) has investigated the resolution of spatial extents based on the diversity of the representative data [17]. Similar techniques have been applied to Land Use / Cover classification [18]. Druaguct et. al [19] find that accuracy of inferences is correlated with the precision of the spatial extent, where smaller resolutions produce better models. Our use-case leverages these techniques to partition satellite imagery for performant co-processing of spatial resolutions.

Several approaches for improving the temporal resolution of satellite imageries have been explored. In [4], authors suggest a spatiotemporal data fusion method to predict surface reflectance with high spatial-temporal resolutions by capturing temporal changes from MODIS images and spatial features such as land cover classification from 30m resolution Landsat-7 NDVI maps and solving the problem as a linear equation using least-squares techniques. The results show a high correlation between synthetic and actual surface reflectance (0.88+ for Landsat bands 1-7). However, the model has been tested using imageries captured on 3 dates each at a temporal spacing of 16 days. As such seasonal and regional variations are not

Satellite	Resolution		Wavelength Range (nm)		
	Temporal (Days)	Spatial (m)	Band 4 (Red)	Band 2 (Green)	Band 2 (Blue)
Sentinel2	5	10	650-680	543-578	458-523
MODIS	1-2	500	620-670	545-565	459-479
NAIP	730	1-2	580-700	480-640	380-580

TABLE I: Spatiotemporal and spectral wavelengths of satellite sensors used in the SATnet model

accounted for. Similar spatial and temporal fusion approaches have been adopted in [7], [8], [20]–[22]. Liu et al use the STARFM model to generate 15m resolution images for the desired timestamp using ASTER and MODIS tiles. In contrast to these models requiring imageries to be ingested in the future and at scheduled intervals, SATnet estimates imageries by capturing changes from only past imageries. Finally, unlike these efforts, SATnet provides broader seasonal and spatial coverage.

Hilker et. el. [21] propose a methodology that derives the spatial change mask at high spatial resolutions using Tasseled Cap indices. Tao et al [23] generate high-resolution images by capturing the spatial distribution of temporal change from image time-series at a lower resolution and projecting onto a high-resolution plane. The accuracy of this model is 68%. Similar approaches proposed in [24], avoids predictions using future timestamp images as inputs. Instead, they propose using an ensemble of multiple forecasting models using Deep Learning algorithms on single NDVI bands. The predicted missing images are fine-tuned by using them as inputs to make predictions for future available timestamps.

More recently, [24] leverages linear regression to model relations between pixels at previous timestamp (T1) and target timestamp (T2) for 300m Sentinel-3 tiles and project on Sentinel 2 previous timestamp available image; thus, producing a high-resolution image from 300m resolution. PSIque model [25] predicts unseen weather conditions using Convolutional Seq2Seq autoencoders and skip-connections based on the past 6 hours of meteorological images captured on an hourly basis at 500m Ground Sample Distance. This model uses a single convLSTM layer with skip connections between encoded and decoded layers. Due to the relatively high MSE, the predicted images lacked clarity compared to other recent approaches. Our method captures temporal variations in features across different spatial regions. These variations between consecutive timestamps across different locations are not uniform due to cloud coverage. We train our models to overcomes such deficiencies (and generalize) to impute for different spatial locations with similar land cover types.

III. Methodology

We now describe our methodology to impute unavailable data using satellite image collections harvested by two different satellite systems (Sentinel-2 and MODIS). We describe both the systems and algorithmic aspects of our methodology.

A. Data Integrating and Staging

Effective data partitioning promotes load-balanced distribution and effective traversal of the dataspace. Satellite imagery introduces additional complications in that sources often differ in spatiotemporal scopes and encoding formats. Reconciling these differences is necessary for practical analysis. Our solution partitions satellite imagery by splitting input images along geohash boundaries [26]. The geohash algorithm is a hierarchical spatial indexing scheme. Regions are defined by character strings, where additional characters partition the region into subregions. Therefore, longer strings represent smaller spatial bounds. We use a distributed hash table (DHT) to distribute the geohash partitioned images. Each geohash is passed through a cryptographic hash function (SHA-256); this facilitates load-balanced data dispersion and subsequent model training workloads. Our DHT key is the geohash (spatial extent) of each partitioned image. Therefore, each spatial extent is co-located, regardless of the satellite system. Consequently, data movements are mitigated when co-processing imagery from multiple sources.

Data Preprocessing Satellite data are reported in a variety of formats (e.g., ZIP, HDF5, netCDF) and spatial reference systems (e.g., latitude/longitude, pseudo, Web Mercator, and sinusoidal). When images are provided in a different spatial reference system (SRS), we perform pixel coordinate transformations to ensure accurate partitioning. Transforming the SRS for each pixel entails processing overheads due to the large number of pixels for a single band. We identify the center points of a given geohash and incrementally adjust pixel boundaries until the minimum bounding region is found.

In Table I, we summarize the spatial and temporal resolution of the three satellite sensors used for training and making inferences by SATnet. Due to the close radiometric resolution for RGB bands of MODIS and Sentinel-2, they possess similar characteristics and are well suited for our application. On the other hand, the NAIP satellite [27] captures spectral bands at slightly lower bandwidths thus resulting in small deviations of band values as compared to Sentinel-2 images captured on the same day at corresponding geolocation.

Image preprocessing, specifically to identify and rectify data occlusions, is vital for subsequent processing tasks. Satellite images with extensive cloud coverage occlude the underlying land information, and this may have an adverse impact on model accuracy.

Data occlusions impede accurate inferences; for example, images with extensive cloud coverage shield the underlying land information. This may adversely impact on the accuracy of generated images. We compute the cloud coverage percentage of each geohash partitioned Sentinel-2 image using multispectral analysis encompassing 10 unique bands. Iterating over the image, we compute the probability of cloud coverage for each pixel using a convolutional neural network based on pixel-wise inputs over a single scene as proposed by SentinelHub [28] under their eo-learn library. We compare these to a threshold value, discretizing the values to true or false, and aggregate the information to compute the overall cloud coverage percentage.

We also regularly reproduce images with many 'fill' values to cope with the differences in spatial bounds between imagery and geohash defined regions. Deep Learning tasks that we will

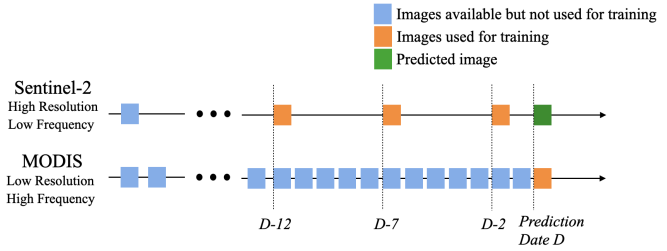

Fig. 1: The retrospective window of the model used for training. For a prediction date D, the model consumes 3 previous dates D - 2, D - 7, and D - 12 high-resolution Sentinel-2 images and a single MODIS image for date D.

describe in the subsequent section, typically benefit from a voluminous training dataset. To this end, we perform image augmentation and retrieve incomplete data from other source images (which share the temporal scope) to produce composite images with low occlusions.

B. Model Structures and hyper parameters [RQ-3]

We have designed SATnet to estimate missing satellite images between scheduled scans. SATnet falls broadly within a class of deep learning networks known as Generative Adversarial Networks (GANs). GANs learn from the pairing of prediction and data distribution. This unique capability allows the predictions/generations to be more realistic especially when sufficient data is available to capture the realistic data distributions. Although the original GAN was developed to generate realistic artificial images, GANs have been widely applied for several different problems such as super resolution [29] and predictions [30]. SATnet leverages fundamental GAN concepts and augments them for the unique characteristics in satellite observations.

SATnet is trained using remotely sensed images collected by two different satellites at different spatial resolutions and temporal (scan) frequencies. In particular, we are working with datasets from Sentinel-2 and MODIS satellites. As depicted in Figure 1, Sentinel-2 provides high resolution (10m) images with a low scanning frequency (every 5 days), and MODIS scans the earth more frequently (every 1-2 days) but at medium resolutions (500m). To predict a high-resolution Sentinel-2 image on day D, when data is unavailable, SATnet takes previous (temporally proximate) observations from Sentinel-2 and the image captured by MODIS on the prediction day. We vary the number of historical images from Sentinel-2 as a part of our model evaluations.

The generator model G learns to map the input features distribution $p_{data}(x)$ to the target distribution and generate realistic image G(z). The goal of this network is to fool the discriminator model D (trained simultaneously with the generator) to distinguish between "real" and "fake" (model generated) images by optimizing the cross-entropy loss between these images [11]. In particular, this is an optimization of the min-max problem represented by Eq.(1) below.

$$\min_G \max_D V(D,G) = \min_G \max_D (E_{x\sim P_{data}(x)}[logD(x)] + E_{z\sim P_z(z)}[log(1-D(G(z)))]) \quad (1)$$

where x and z represents the input high and low resolution data respectively.

Unlike images used in traditional computer vision [4], [8], [20], [21], satellites capture images of the same location repeatedly over time. This results in two distinctive aspects that must be considered: we characterize them as slow-evolving components and fast-evolving components. Many structural and topographical components such as road systems, buildings, or the shape of a lake do not change very often and are characterized as slow-evolving components. Meanwhile, bands that capture reflectance from crops or plants may change based on the plant's lifecycle or weather change; these are characterized as fast-evolving components. Our model accounts for both slow-evolving and fast-evolving components.

Generator [RQ-2]. As shown in Figure 2a, the model utilizes the previous n (> 1) Sentinel-2 images along with the temporally proximate MODIS image captured at the queried timestamp. These n Sentinel-2 tiles are fed to the Bidirectional ConvLSTM layer, followed by a downscaling layer to capture low-level features from the input images. Alongside this, the model upscales the input MODIS tile (16,16,3). At each block-level, a skip connection is added to the upscaled MODIS features to capture slow-evolving components such as road networks, agricultural structures, patterns, and shapes of objects. Accounting for these features enables the model to preserve the geographical properties of the particular region. To avoid model overfitting, we leverage dropout regularization within the generator models. Dropouts deactivate some neurons randomly during training, thus avoiding any overfitting of the model.

LSTMs have been widely acknowledged as an effective network to address ordered sequence learning problems based on the assumption that the previous state affects future states [31]. However, unlike traditional sequence learning problems, spectral channels within the sequence are correlated with each other [32]. Direct application of the method destroys the two-dimensional structure of images, leading to the loss of spatial relationships. To address this deficiency, we leverage Bi-CLSTM [12] as our generator model that takes the sequence of input images incrementally with a forward and a backward sequence. Bi-CLSTM allows SATnet to take local image patch as inputs so that it uses a locally-connected, weight sharing structure to extract spatial features as depicted in Figure 3.

The operation inside the Bi-CLSTM memory cell can be represented using the following equation set (2), where b_f, b_i, b_o represents the biases and W_{Xf}, W_{hf}.. are the weight matrices that are updated during model training. $sigma$ represents the logistic sigmoid activation function for the Convolution LSTM cell gates.

$$\begin{aligned} f_T &= \sigma(W_{Xf} * X_T + W_{hf} * h_{T-1} + b_f) \\ i_T &= \sigma(W_{Xi} * X_T + W_{hi} * h_{T-1} + b_i) \\ \bar{c}_T &= \tanh(W_{Xc} * X_T + W_{hc} * h_{T-1}) \\ o_T &= \sigma(W_{Xo} * X_T + W_{ho} * h_{T-1} + b_o) \end{aligned} \quad (2)$$

The cell consists of a forget gate, an input gate, and an

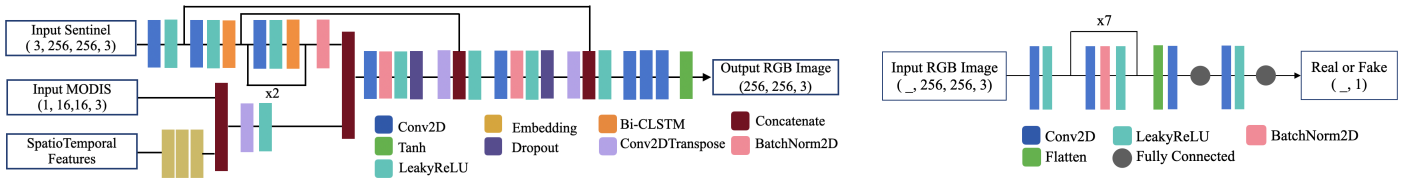

Fig. 2: (a) The architecture of the SATnet Generator. The generator is trained with a MODIS image acquired at a target timestamp (+/- 1 day) and 3 temporally proximate Sentinel-2 images. (b) The architecture of the SATnet GAN discriminator.

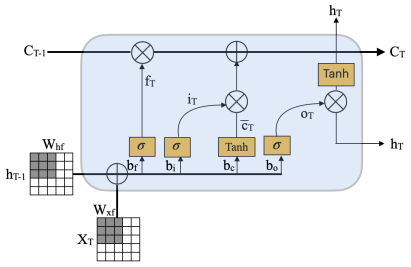

Fig. 3: The Architecture of Convolution LSTM cell.

output gate, that operates on a patch of features extracted from input images. The forget gate f_T controls the information that needs to be retained in the cell. The input gate i_T decides information that needs to be updated based on candidate value $\bar{c}_T$ that further controls the flow of new value in the cell. Using cell state C_T and output gate o_T, the memory cell controls the output information of the unit.

$$\begin{aligned} C_T &= f_T \cdot C_{T-1} + i_T \cdot \bar{c}_T \\ h_T &= o_T \cdot \tanh(C_T) \end{aligned} \tag{3}$$

SATnet also benefits from the network property of LSTMs with its demonstrably low sensitivity irregular time intervals between inputs [31]; this property is well suited for our training set. The prediction date D can be 1-4 days in the future from the most recent Sentinel-2 imagery. Occlusions due to heavy cloud coverage result in uneven gaps between the imageries when selecting the training dataset.

To capture relationships between fast-evolving components, SATnet integrates metadata such as the geohash and seasonal information (e.g. the week of the year of the target timestamp). This ancillary information accelerates training by finding local patterns based on seasonality for a given location. Often neighboring geohashes comprise similar land cover types, thus helping the model learn geographical patterns. The ancillary information is utilized alongside MODIS image features in the initial stage of the network.

Discriminator The goal of a discriminator network in GAN is to classify the generated images as real or fake [10]. If the inferred image is classified as fake, then the generator model is not performing well and thus optimizes the weights from the loss value of the discriminator model. As depicted in Figure 2b, the discriminator comprises a block of convolution layers followed by a batch normalization layer and Leaky Relu activation function. This block is repeated to extract important features from the input images and perform classification. The convolutional operations are performed to reduce the image size gradually after which fully connected layers are used to predict the probability of the input sample.

Model Variation	Loss Combination		Testing Metrics		
	Adversial Loss	Content Loss	PSNR	MSE	SSIM
1	Yes	No	30.13	0.0056	0.89
2	Yes	Yes	31.82	0.00096	0.93
3	No	Yes	28.73	0.0099	0.8552

TABLE II: SATnet model performance when incorporating different loss values. SATnet performs best with weighted adversarial and content loss for weight updates.

Incorporating content loss SATnet is trained using the discriminator's adversarial loss, which is calculated using a cross-entropy loss function. Discriminator loss represents the probability estimate of how realistic the generated image is. Aside from discriminator loss, we combine the content loss, which is the mean squared error between feature maps extracted using a pre-trained VGG19 model from the inferred image and the target image. Using VGG features as content loss helps in generating a better quality image instead of optimizing the raw pixel values, we minimize the errors between low-level features such as edges and textures. This argument is further substantiated by experiment results shown in Table II, where we report the performance of the SATnet model with variations in the integrated training loss. A model trained without an adversarial loss (Model Variation 1) results in a 10% decrease in overall accuracy. A model (Model Variation 3) trained with MSE loss function in Eq.(4) and adversarial loss results in slightly increased errors. The best model performance is accomplished by the inclusion of both content loss and adversarial loss while training (Model Variation 2).

$$MSE = \frac{1}{MN}\sum_{n=1}^{N}\sum_{m=1}^{M}[x_{(n,m)} - y_{(n,m)}]^2 \tag{4}$$

$$MAE = \frac{1}{MN}\sum_{n=1}^{N}\sum_{m=1}^{M}|x_{(n,m)} - y_{(n,m)}| \tag{5}$$

$$L = \frac{1}{MN}\sum_{n=1}^{N}\sum_{m=1}^{M} log(cosh(x_{(n,m)} - y_{(n,m)})) \tag{6}$$

$$SSIM = l(x,y)^{\alpha} \cdot c(x,y)^{\beta} \cdot s(x,y)^{\gamma} \tag{7}$$

$$L_{mix} = (1-\alpha) \cdot MSE + \alpha \cdot SSIM \tag{8}$$

where M, N are number of rows and columns in predicted image x and target image y. Functions l, c, and s measure the luminance, contrast, and structure comparisons between inputs images respectively. Parameters α, β and γ represents the relative importance of these three factors (l, c, and s).

Test Metrics	Training Loss				
	MSE	**Logcosh**	**Huber**	**SSIM & MSE**	**MAE**
PSNR	28.97	27.87	31.82	30.29	28.34
MSE	0.0098	0.0114	0.00096	0.0043	0.0095
SSIM	0.8164	0.7938	0.93	0.9037	0.8588

TABLE III: Model performance while training with various cost functions as content loss. Best performance achieved when weights are updated with Huber loss (PSNR 31.82).

To estimate and update the weights of the model's parameters, we experimented with multiple cost functions as the content loss: MSE (Mean squared error) (4), MAE (Mean Absolute Error) (5), Huber loss (9), logcosh (6), and the weighted average of SSIM and MSE (8).

From Table III, the best results are obtained with Huber loss as the cost function with δ fixed at 0.2. Using Logcosh results in the worst performance of the model. Using the Huber cost function results in absolute errors becoming quadratic for when the errors are small. Absolute errors are usually less sensitive to outliers; however, Huber loss helps in achieving combined properties of the MSE and MAE, where MSE leads to more precise minima, while MAE leads to precise minima. This makes Huber loss most suitable for our context where certain training samples have cloud-covered regions.

$$L_{huber} = \begin{cases} \frac{1}{2}(y-x)^2, & \text{if } |y-x| \leq \delta \\ \delta|y-x| - \frac{1}{2}\delta^2, & \text{otherwise} \end{cases} \quad (9)$$

,where $\delta \in \mathbb{R}^+$ is a hyperparameter.

C. Distributed Model Training

Distributed training consumes Sentinel-2 and MODIS images that are stored locally on each machine within the cluster. We retrieve Sentinel-2 images from the underlying storage system at a geohash length of 5, which maps to a spatial extent of roughly 100km x 100km region centered on Fresno, CA. Images that have cloud coverage of more than 40% are excluded from the training set. The cloud coverage is estimated as a part of data preprocessing (section III-A).

To train SATnet, we leverage a set of metadata correlated with images generated by the underlying storage system, such as information about data compression, the timestamp at which the image was acquired by the satellite, and a corresponding geohash. When training models, we pair Sentinel-2 and MODIS images by querying for the particular region at the timestamp (+/- 1 day) of the corresponding Sentinel-2 images. The coverage and geospatial locations of these images are aligned and mapped by sharing the same geohash (with the same length). We also account for temporal information to detect seasonal change. In this study, we use the normalized week number (1 through 52) and the season of the year when the satellite image was captured. Lastly, we convert this spatial extent information to generate numerical encodings.

Since SATnet comprises LSTM as a main component of the generator, the training sequences are generated by sliding through the collection. We exclude timestamps without corresponding temporally proximate MODIS image and also images where the Sentinel-2 image in the sequence breaches the *retrospective window* threshold. To preserve temporal proximity across the sequence of images, the retrospective window constrains how far back in the past, from a particular timestamp, the algorithm is allowed to retrieve data from. For example, if SATnet takes 3 Sentinel images, the retrospective window is 24 days; the earliest image cannot be older than 24 days from the imputation time. For 2 and 4 image counts, we apply the retrospective window threshold to 16 days and 32 days respectively.

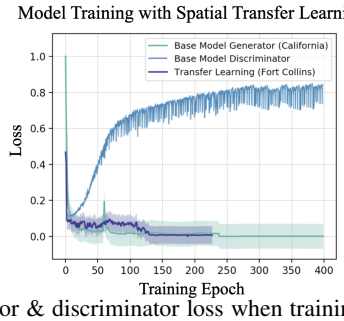

Fig. 4: Generator & discriminator loss when training base model and performing transfer learning at other spatial regions.

D. Spatial Transfer Learning

Training a model specialized over a certain region not only reduces the model storage cost but can also be leveraged for making inferences over similar spatial regions. With a different distribution of landcover types and climatic and weather conditions, trends learned over slow-evolving objects remain similar across regions. The fast-evolving features can be further learned by fine-tuning the model specialized over a particular region.

To test the quality of inferences made by SATnet models trained on the Fresno region in California while making inferences over a 20km x 20km area in Fort Collins, Colorado. Although the land coverage of both regions has similar footprints, there are vast differences in climatic conditions.

Figure 4 depicts the training loss of the base model on California satellite observations and model learning from this base model (Transfer Learning). Further, we report the discriminator loss while training the base model. The model quickly learns to generate the slow-evolving spatial features of a given location, after which it fine-tunes to seasonal changes i.e., fast-evolving components. The increasing discriminator error over training epochs verifies that the generator network is generating realistic images that pass muster with the discriminator network used in the SATnet model.

E. Support for dynamic, query-driven imputation

Our system supports distributed query-driven imputations. Queries are defined by spatiotemporal bounds, providing latitude and longitude coordinates and a timestamp. The query output is a snapshot image of the specified spatial bounds at the desired timestamp with the spatial resolution of Sentinel-2. The image is either an actual image(if it exists) or an imputed image (if it does not exist). We evaluate queries in a distributed fashion over the DHT nodes. Any node may serve as a conduit for query evaluation and an instance of the trained model is installed at each node.

To evaluate the query, the system first identifies the subset of nodes that need to be included in the evaluation. We decompose the query into a collection of associated geohash regions and locate the nodes that are responsible for the corresponding geohashes. Then, our system collects necessary images from the DHT. Each node determines the source of our requested image. If a matching image is stored in the system, the original image will be returned. Otherwise, we impute an image using a local SATnet with temporally proximate Sentinel-2 and MODIS data. If data sparsity issues preclude completion of the imputation process, the DHT node reports the failure to the conduit node.

Finally, retrieved images are merged into a single image to be returned to the user. The partial images are profiled for properties such as dimensionality, pixel coordinate transformations, and spatial reference systems. Then, the data is aggregated into a composite image, comprising a mix of original and imputed, synthetic imagery. This step also involves trimming edges of images, because the queried spatial bounds often do not align with geohash bounds, and rectifying image resolutions. Any unavailable tile is rendered using 'fill' values (visualized as black) reflecting the unavailability of the data.

IV. Performance Evaluation

We performed several experiments to assess our methodology. All the experiments were performed on a cluster of 45 nodes (Xeon E5-2620, 64 GB Memory), each with a single Quadro P2200 GPU (5GB of memory) with 128 cores. The model was implemented using Tensorflow-GPU 2.3. The input satellite images and numerical ancillary information were normalized between the range [-1 to 1]. The learning rate for the generator was fixed at 0.0004 with a batch size of 4 with a learning rate schedular that decays the rate by 0.2 after every 100 epochs.

A. Datasets

Table I outlines the datasets we have leveraged in this work. The global resolution Sentinel-2 imagery is provided by the Copernicus Programme as part of the European Space Agency. Their twin satellite (ie. Sentinel-2A and Sentinel-2B) capture multi-spectral imagery comprising 13 visible, near-infrared, and shortwave infrared bands. NASA's MODIS captures 36 bands with spatial resolutions ranging from 250m to 1km. The low spatial and high temporal resolutions in this data make it particularly useful for tracking changes in the landscape over time. NAIP imagery, provided by USDA's Farm Service Agency has a high spatial resolution (1m) and is imagery acquired during the growing seasons in the Continental US. These images are temporally sparse, compiled every 3 years since 2009 and every 5 years before that. Our evaluation focuses on the red, green, and blue bands provided by these data, leveraging the varying spatial and temporal resolutions to produce fast, accurate imputations.

B. Cluster Setup

We train SATnet over the dispersed dataset (See section III-A) using Tensorflow. The distributed training is based upon Baidu ring-all reduce strategy and is supported by the open-source framework Horovod [33]. Horovod leverages NVIDIA's NCCL library for optimized collective computations over multiple machines. This overcomes the limitation of having a centralized parameter server where all the nodes have to report the weights and gradients to the parameter server resulting in a bottleneck. Using the ring-all reduce approach, every node in the cluster communicates with only two other nodes for $2 * (N - 1)$ pairwise communications in a cluster of N nodes. The time taken to perform an average gradient update remains constant with an increasing number of nodes in the cluster, making model training scalable with increased nodes in the cluster, this can be supported by Table VI.

The models are trained on a 100 km x 100 km region covering the Fresno region in California. This comprises imageries from diverse landcovers such as agricultural farms, mountains, and urban regions. The training datasets temporal coverage is from January 2018 to June 2019 accounting for about 20 GB of data for training. We consider the model as having converged if there is no error drop for 50 epochs. The centralized trained model is stored locally on each machine in the cluster for making imputations and performing evaluations. To measure performance, we calculate the commonly used image quality measure Peak-to-Signal-Noise ratio (PSNR), MSE in Eq.(4), and SSIM in Eq.(7). PSNR measures the quality of the image by calculating the ratio between the maximum possible value of the original image with corruption observed in the estimated image. On another hand Structural Similarity index (SSIM), measures the difference, in contrast, luminance and structure using the sliding window over images.

C. Model Parameters [RQ-1]

In Figure 5a, we contrast the performance of models trained with a retrospective window sizes of 1, 2, 3, 4 and 5 for Sentinel-2 inputs. We measure the performance in terms of the accuracy of inferences, the time taken to infer, and the model size represented by the bubble size. The retrospective window size of 1 results in the smallest model (132 Mb) and infers fastest (5.8 ms) as expected; however, using the most recent image does not incorporate any trend information over time and leads to worse accuracy of 25.1 PSNR. In general, the size of the model and inference time increases steadily with increasing sequence length. The model with a window size of 3 performs the best: we achieve the highest PSNR of 31.82 at decent model inference time (17.9 ms).

In the next set of experiments, we perform a detailed examination of the effect of cloud-covered regions in the input time series sequence while training and making inferences from the model. Figure 6a depicts the model sensitivity towards input images contaminated by high cloud coverage. The model with a retrospective window of 3 outperforms the other models, with the best performance (PSNR of 31.38) on the cloud threshold of 10% on training and testing input images. Errors are drastically reduced by an average of 47% when the window size is increased from 2 to 3. As the cloud threshold increases to 50%, model performance decreases by 5.2%, 4.9%, 3.82%, and 2.20% for a retrospective window size of 2, 3, 4, and 5 respectively. A larger window size results in the reduction

Retro-spective Window Size(Sn)	Cloud Coverage Threshold (in percentage)														
	10%			20%			30%			40%			50%		
	Total Images	Testing Errors		Total Images	Testing Errors		Total Images	Testing Errors		Total Images	Testing Errors		Total Images	Testing Errors	
	Test	mse	std	Test	mse	std	Test	mse	std	Test	mse	std	Test	mse	std
Sn = 2	688	0.00173	0.0005	1684	0.00201	0.0007	2554	0.00260	0.0025	3252	0.00292	0.0017	3826	0.00351	0.0041
Sn = 3	538	0.00091	0.0005	1276	0.00112	0.0007	2103	0.00141	0.0010	2478	0.00141	0.0013	2968	0.00189	0.0015
Sn = 4	256	0.00117	0.0003	644	0.00127	0.0005	924	0.00150	0.0010	1170	0.00177	0.0008	1340	0.00273	0.0004
Sn = 5	374	0.00104	0.0005	1022	0.00124	0.0003	1601	0.00160	0.0007	2118	0.00179	0.0007	2544	0.00240	0.0006

TABLE IV: Number of Training and Testing Samples with varying cloud coverage and retrospective window size

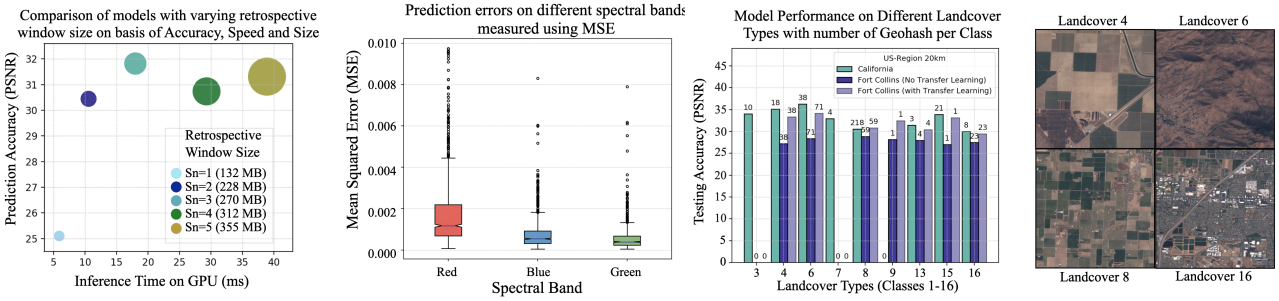

Fig. 5: (a) Contrasting computation requirements and testing accuracy by changing the size of retrospective window during model training. Size of bubble represents the model size. (b) SATnet testing errors for each band with a window size of 3. Red band has the highest mean MSE of 0.0011 with 25% of the samples having an MSE above 0.002. (c) Performance over various landcover types for the base model trained with data from Fresno California; base model inferences are made over a region in Colorado and after transfer learning. (d) Base model inferences made for the dominant landcover classes.

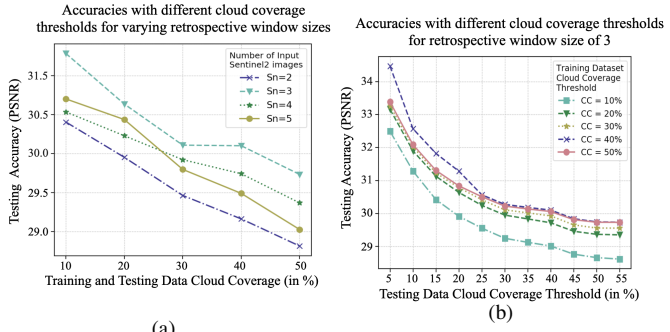

Fig. 6: (a) Decreasing cloud coverage in training data significantly increases accuracy across all retrospective window sizes. At the window size of 3, highest accuracy is achieved. (b) Training SATnet with a fixed window size of 3 and varying the cloud coverage on train and test samples.

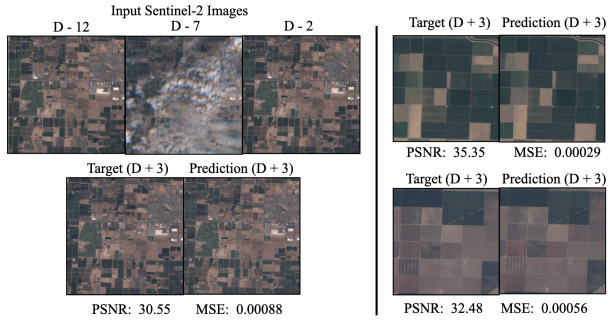

Fig. 7: Representative examples of 10m resolution satellite images generated by our SATnet model.

of training samples as reported in Table IV. Since capturing consecutive images with a specified cloud threshold for a longer sequence length is challenging, the number of testing samples decreases as the retrospective window size increases. We report errors with standard deviations; with Sn = 4, the model overfits as input sample size considerably reduces.

Models	Testing Metrics		
	PSNR	MSE	SSIM
ConvLSTM	26.18	0.008404	0.72
Dynamic Network Filter	28.32	0.004353	0.83
TrajGRU	30.65	0.001163	0.91
SATnet (Ours)	31.82	0.000969	0.93

TABLE V: Contrasting performance for different time series prediction models. SATnet provides the best PSNR.

D. Comparison with ConvLSTM, TrajGRU and DFN

Next, we contrast SATnet with a ConvLSTM model, and specialized time series prediction models such as Dynamic Network Filter (DFN) [34] and TrajGRU [35]. The dynamic filter network prediction model is widely used in video and stream prediction such as on moving MNIST data to predict the spatial location of rotating digits in the next time frame. The input image acts as the condition for setting up the number of filters on-the-fly while training the model instead of fixed pre-defined filter values.

TrajGRU is developed on top of GRUs and ConvLSTMs to predict location-variant transformations. Our model is based upon Bi-CLSTM, which is highly efficient in capturing the spatiotemporal relations with location-invariant input sequences. On other hand, TrajGRU is more specific for input sequences, where the object position is changed over time such as hourly cloud prediction, moving MNIST prediction, or precipitation forecast.

We also compared our model performance with a simpler network consisting of three stacked ConvLSTMs layers and convolutional layers, for making inferences. All three models are provided with Sentinel-2 images of a retrospective window size of 3 with a maximum cloud coverage threshold of

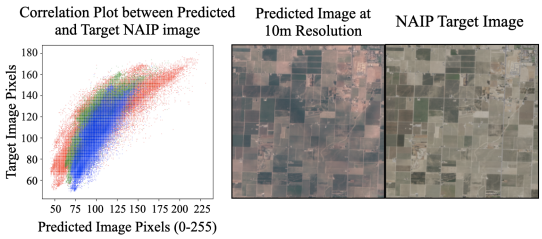

Fig. 8: SATnet performance while making a high-resolution imputation on geohash 9q6z0 for timestamp D, where temporally proximate Sentinel-2 images for dates D-2, D-7, D-12 and MODIS for D are fed as inputs.

40%. Table V shows that our model outperforms the other model variations. The stacked ConvLSTM model performs worst with a PSNR of 26.18 and is inefficient in making any useful inferences. TrajGRU performs comparably well with a PSNR of 30.65. For our problem set, the location of urban regions, road networks, and farms remains constant over time. The only moving component is cloud and snow cover, which drastically change over a 5-day interval. In summary, our model outperforms DFN and TrajGRU which are more specialized for location-variant inputs.

To measure SATnet performance for prediction within 5 days of interval, we compare the image quality by generating a correlation plot (see Figure 8), between RGB band values of both inferred and target image collected from NAIP satellite. The pixel values in the first plot, show the linear relation between the band values. Blue and green band pixels are less dispersed showing low errors, while the red band forms scattered clouds. This error can be seen in the samples, where NAIP image pixel intensities are slightly off from Sentinel-2. While making inferences for timestamp that is 5 days apart from the most recent available Sentinel-2 image, we observe low errors of 0.0005 and 0.0007 in green and blue band respectively. However, the red band has high MSE of 0.001 (see Figure 5b).

E. Transfer Learning

Spatial transfer learning can help avoid cold-start training of the base model on similar regions by leveraging high-level presentational features of that region from the pre-trained model. Further, the spectral properties of bands belonging to the same landcover have similar pattern changes over time. We explored how the model performs on satellite images with the same landcover types but different locations before and after performing transfer learning. Using the National Land Cover Database (NLCD) [36], we classify each geohash into 16 possible classes. These classes broadly cover various forest types, grasslands, crops, etc. The majority of Colorado regions are classified into classes 4, 6, and 8 that overlaps with the California region and well suited for performing transfer learning. As shown in Figure 5c, on classes 6 and 8 majorly covering the agricultural and mountain regions over Colorado, the base model accuracies are slightly higher compared to other classes. However, after performing transfer learning, land covers 4, 9, and 15 inferences show drastic improvement in accuracy. The overall testing accuracy of the model before

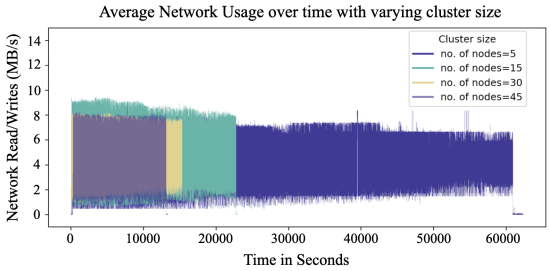

Fig. 9: Average Network I/O incurred for initial 100 epochs when training SATnet with varying cluster size.

Cluster Size	Average Network I/O	Total Network I/O	Total Training Time
n = 5	4.085 MB/sec	254.68 GB	48 hrs
n = 15	14.12 MB/sec	323.25 GB	15.6 hrs
n = 30	27.96 MB/sec	428.31 GB	12.4 hrs
n = 45	37.51 MB/sec	500.79 GB	10.5 hrs

TABLE VI: Computation cost for training SATnet on different cluster sizes. Total network I/O increases with increase in cluster size, however at considerably lower training duration.

and after performing transfer learning is 28.3 and 31.1 PSNR. Colorado areas are also affected by a high amount of snow, which can affect the overall testing accuracies.

While the transfer learning model converges faster when compared to the base model, the overall training loss is slightly higher for the Colorado region. The Fort Collins region is covered by snow more often than the Fresno region, the overall accuracy is lower due to the uneven snow coverage in input sample images. Images predicted by the SATnet for geohashes belonging to the prominent classes is shown in Figure 5d.

F. Distributed Training Cluster Size

We performed experiments to measure the cost of training SATnet with a varying number of machines in a distributed cluster. We trained the model by fixing the number of nodes to 5, 15, 30, and 45, where training data spread over 45 machines is redistributed equally to machines used in the cluster. This avoids any network overhead incurred by transferring the data across machines and therefore results in reading data locally.

Network I/O is incurred during distributed training to share the model weights and gradients across multiple machines and perform average weight update after each training epoch. Figure 9 depicts the network I/O cost during the training of SATnet with varying cluster sizes for 100 epochs. Leveraging the power of ring-all reduce distributed strategy, we can see that, average network I/O incurred by each machine does not exponentially grows sub-linearly with increasing cluster size.

G. Distributed Query Evaluation

We also profiled the image imputation process when the entire system is subject to cold-start using both PyTorch and TensorFlow. During imputations, we retrieve and process temporally proximate Sentinel-2 and MODIS imagery to generate accurate, synthetic data. Here, we report cold-start times, where the entire system is operating on non-cached imagery: as a result, the imputation process involves expensive disk I/O to retrieve images. On a given machine, we achieve throughputs of a 10 and 11 images per second using PyTorch v1.7 and Tensorflow v2.5 respectively. Notably, our imputation

process scales linearly with additional machines; during cold-start benchmarks, over a cluster of 50 machines, we are able to generate synthetic imagery at over 500 images per second. The Tensorflow framework marginally outperforms PyTorch for our use-case, because GPU memory is managed a little more efficiently in Tensorflow enabling larger simultaneous batch evaluations. To ensure effective GPU utilizations, we avoid blocking during disk and network I/O. We remove the aforementioned operations from the critical path by implementing a data pipeline and apportioning functionality over multiple stages to ensure non-blocking I/O operations.

V. Conclusions and Future Work

We described a novel methodology to estimate high-resolution satellite images at arbitrary spatial and temporal scopes. Our methodology leverages deep learning algorithms to effectively learn temporal patterns in remote sensing data.

RQ-1 The SATnet model training is boosted by our underlying distributed storage system which allows us to reduce training times drastically by leveraging data locality during training and avoiding network I/O. We trained the SATnet model 4x faster by utilizing the distributed cluster without incurring drops in accuracy.

RQ-2 We reconstruct the high-resolution spatial characteristics of a spatial region using our GAN architecture. The model captures spatial patterns and accounts for temporal changes; this is reflected in our superior PSNR accuracy of 31.82.

RQ-3 By utilizing high-frequency scans from MODIS, we capture patterns in slow-evolving features. By using high-resolution Sentinel-2 data, we preserve spatial characteristics of the given region.

As future work, we plan to apply the proposed approach to other remote sensing data, such as filling Landsat images with temporal resolution on 16-17 days at 30m spatial resolution using the PROBA-V satellite, which scans the earth every two days at 300m resolution. The approach can further applied to various bands such as NIR and SWIR that are primarily used for calculating vegetation indexes over landcovers.

Acknowledgment

This research was supported by the National Science Foundation [OAC-1931363, ACI-1553685] and the National Institute of Food & Agriculture [COL0-FACT-2019].

References

[1] B. L. Markham et al. Landsat 8: status and on-orbit performance. In *Sensors, Systems, and Next-Generation Satellites XIX*, volume 9639. International Society for Optics and Photonics, 2015.

[2] M. Drusch et al. Sentinel-2: Esa's optical high-resolution mission for gmes operational services. *Remote sensing of Environment*, 120, 2012.

[3] C. Schaaf and Z. Wang. Mcd43a4: Modis/terra+ aqua brdf/albedo nadir brdf l3 global-500m. *NASA EOSDIS Land Processes DAAC*, 2015.

[4] M. Wu et al. Use of modis and landsat time series data to generate high-resolution temporal synthetic landsat data using a spatial and temporal reflectance fusion model. *Journal of Applied Remote Sensing*, 6, 2012.

[5] H. Shen et al. Missing information reconstruction of remote sensing data. *IEEE Geoscience and Remote Sensing Magazine*, 3, 2015.

[6] J. L. Crespo et al. A new image prediction model based on spatio-temporal techniques. *The Visual Computer*, 23(6), 2007.

[7] Q. Wang and P. M. Atkinson. Spatio-temporal fusion for daily sentinel-2 images. *Remote Sensing of Environment*, 204, 2018.

[8] F. Gao et al. On the blending of the landsat and modis surface reflectance: Predicting daily landsat surface reflectance. *IEEE Transactions on Geoscience and Remote sensing*, 44(8), 2006.

[9] H. K. Zhang et al. A generalization of spatial and temporal fusion methods for remotely sensed surface parameters. *International Journal of Remote Sensing*, 36(17), 2015.

[10] C. Ledig et al. Photo-realistic single image super-resolution using a generative adversarial network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017.

[11] I. Goodfellow et al. Generative adversarial nets. In *Advances in neural information processing systems*, 2014.

[12] Q. Liu et al. Bidirectional-convolutional lstm based spectral-spatial feature learning for hyperspectral image classification. *Remote Sensing*, 9(12), 2017.

[13] S. Lang et al. Geobia achievements and spatial opportunities in the era of big earth observation data. *ISPRS*, 8(11), 2019.

[14] L. Hou et al. High resolution medical image analysis with spatial partitioning. *arXiv preprint arXiv:1909.03108*, 2019.

[15] L. Xu et al. Farmland extraction from high spatial resolution remote sensing images based on stratified scale pre-estimation. *Remote Sensing*, 11(2), 2019.

[16] H. Bazzi et al. Mapping irrigated areas using sentinel-1 time series in catalonia, spain. *Remote Sensing*, 11(15), 2019.

[17] S. Georganos et al. Scale matters: Spatially partitioned unsupervised segmentation parameter optimization for large and heterogeneous satellite images. *Remote Sensing*, 10(9), 2018.

[18] U. Maulik and S. Bandyopadhyay. Fuzzy partitioning using a real-coded variable-length genetic algorithm for pixel classification. *IEEE Transactions on geoscience and remote sensing*, 41(5), 2003.

[19] L. Drăguţ et al. Sensitivity of multiresolution segmentation to spatial extent. *International Journal of Applied Earth Observation and Geoinformation*, 81, 2019.

[20] H. Liu and Q. Weng. Enhancing temporal resolution of satellite imagery for public health studies: A case study of west nile virus outbreak in los angeles in 2007. *Remote Sensing of Environment*, 117, 2012.

[21] T. Hilker et al. A new data fusion model for high spatial-and temporal-resolution mapping of forest disturbance based on landsat and modis. *Remote Sensing of Environment*, 113(8), 2009.

[22] X. Zhou et al. Developing a fused vegetation temperature condition index for drought monitoring at field scales using sentinel-2 and modis imagery. *Computers and Electronics in Agriculture*, 168, 2020.

[23] T. Guo. A method for generating high-resolution satellite image time series. In *Image and Signal Processing for Remote Sensing XX*, volume 9244. International Society for Optics and Photonics, 2014.

[24] M. Das and S. K. Ghosh. A deep-learning-based forecasting ensemble to predict missing data for remote sensing analysis. *IEEE Journal in Applied Earth Observations and Remote Sensing*, 10(12), 2017.

[25] S. Hong et al. Psique: Next sequence prediction of satellite images using a convolutional sequence-to-sequence network. *arXiv:1711.10644*, 2017.

[26] G. Niemeyer. Geohash. http://www.geohash.org/, 1999.

[27] *NAIP Imagery*, 2004. https://www.fsa.usda.gov/programs-and-services/aerial-photography/imagery-programs/naip-imagery/.

[28] L. Sinergise Laboratory for geographical information systems. Sentinel hub member of euro data cube.

[29] X. Wang et al. Esrgan: Enhanced super-resolution generative adversarial networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

[30] P. Singh and N. Komodakis. Cloud-gan: Cloud removal for sentinel-2 imagery using a cyclic consistent generative adversarial networks. In *IEEE International Geoscience and Remote Sensing Symposium*, 2018.

[31] F. A. Gers et al. Applying lstm to time series predictable through time-window approaches. In *Neural Nets WIRN Vietri-01*. Springer, 2002.

[32] N. A. Al Muntshry. Evaluating the effectiveness of multi-spectral remote sensing data for lithological mapping in arid regions. 2011.

[33] A. Sergeev and M. Del Balso. Horovod: fast and easy distributed deep learning in tensorflow. *arXiv preprint arXiv:1802.05799*, 2018.

[34] X. Jia et al. Dynamic filter networks. In *Advances in neural information processing systems*, 2016.

[35] X. Shi et al. Deep learning for precipitation nowcasting. In *Advances in neural information processing systems*, 2017.

[36] C. G. Homer et al. The national land cover database. Technical report, Reston, VA, 2012.