Tile2Vec: Unsupervised Representation Learning for Spatially Distributed Data

Neal Jean,[1,2] Sherrie Wang,[3,4] Anshul Samar,[1] George Azzari,[4] David Lobell,[4] Stefano Ermon[1]
[1]Department of Computer Science, Stanford University, Stanford, CA 94305
[2]Department of Electrical Engineering, Stanford University, Stanford, CA 94305
[3]Institute of Computational and Mathematical Engineering, Stanford University, Stanford, CA 94305
[4]Department of Earth System Science, Stanford University, Stanford, CA 94305
{nealjean, sherwang, asamar, gazzari, dlobell}@stanford.edu, ermon@cs.stanford.edu

Abstract

Geospatial analysis lacks methods like the word vector representations and pre-trained networks that significantly boost performance across a wide range of natural language and computer vision tasks. To fill this gap, we introduce Tile2Vec, an unsupervised representation learning algorithm that extends the distributional hypothesis from natural language — words appearing in similar contexts tend to have similar meanings — to spatially distributed data. We demonstrate empirically that Tile2Vec learns semantically meaningful representations for both image and non-image datasets. Our learned representations significantly improve performance in downstream classification tasks and, similarly to word vectors, allow visual analogies to be obtained via simple arithmetic in the latent space.

1 Introduction

Remote sensing, the measurement of the Earth's surface through aircraft- or satellite-based sensors, is becoming increasingly important to many applications, including land use monitoring, precision agriculture, and military intelligence (Foody 2003; Mulla 2013; Oshri et al. 2018). Combined with recent advances in deep learning and computer vision (Krizhevsky, Sutskever, and Hinton 2012; He et al. 2016), there is enormous potential for monitoring global issues through the automated analysis of remote sensing and other geospatial data streams. However, recent successes in machine learning have largely relied on supervised learning techniques and the availability of very large annotated datasets. Remote sensing provides a huge supply of data, but many downstream tasks of interest are constrained by a lack of labels.

The research community has developed a number of techniques to mitigate the need for labeled data. Often, the key underlying idea is to find a low-dimensional *representation* of the data that is more suitable for downstream machine learning tasks. In many NLP applications, pre-trained word vectors have led to dramatic performance improvements. In computer vision, pre-training on ImageNet is a *de facto* standard that drastically reduces the amount of training data needed for new tasks. Existing techniques, however, are not suitable for remote sensing data that, while superficially resembling natural images, have unique characteristics that require new methodologies. Unlike natural images — object-centric, two-dimensional depictions of three-dimensional scenes — remote sensing images are taken from a bird's eye perspective, and they are also often *multi-spectral*. These differences present both challenges and opportunities. On one hand, models pre-trained on ImageNet do not transfer well and cannot take advantage of additional spectral bands (Xie et al. 2016). On the other, there are fewer occlusions, permutations of object placement, and changes of scale to contend with — this spatial coherence provides a powerful signal for learning representations.

Our main assumption is that image tiles that are geographic neighbors (i.e., close spatially) should have similar semantics and therefore representations, while tiles far apart are likely to have dissimilar semantics and should therefore have dissimilar representations. This is akin to the *distributional hypothesis* used to construct word vector representations in natural language: words that appear in similar contexts should have similar meanings. The main computational (and statistical) challenge is that image patches are themselves complex, high-dimensional vectors, unlike words.

In this paper, we propose Tile2Vec, a method for learning compressed yet informative representations from unlabeled remote sensing data. We evaluate our algorithm on a wide range of remote sensing datasets and find that it generalizes across data modalities, with stable training and robustness to hyperparameter choices. On a difficult land use classification task, our learned representations outperform other unsupervised features and even exceed the performance of supervised models trained on large labeled training sets. Tile2Vec learns a meaningful embedding space, demonstrated through visual query by example, latent space interpolation, and visual analogy experiments. Finally, we apply Tile2Vec to the non-image task of predicting country health indices from economic data, suggesting that real-world applications of Tile2Vec may extend to domains beyond remote sensing.

2 Tile2Vec

For clarity, in this section we focus on the application of Tile2Vec to remotely sensed image datasets. The extension to non-image spatial data is straightforward, and we revisit this setting in Section 4.4.

2.1 Distributional semantics

The distributional hypothesis in linguistics is the idea that "a word is characterized by the company it keeps". In NLP, algorithms like Word2vec and GloVe leverage this assumption to learn continuous representations that capture the nuanced meanings of huge vocabularies of words. The strategy is to build a co-occurrence matrix and solve an implicit matrix factorization problem, learning a low-rank approximation where words that appear in similar contexts have similar representations (Levy and Goldberg 2014; Pennington, Socher, and Manning 2014; Mikolov et al. 2013b).

To extend these ideas to geospatial data, we need to answer the following questions:

- What is the right atomic unit, i.e., the equivalent of individual words in NLP?
- What is the right notion of context?

For atomic units, we propose to learn representations at the level of remote sensing *tiles*, a generalization of image patches to multispectral data. This introduces new challenges as tiles are high-dimensional objects — computations on co-occurrence tensors of tiles would quickly become intractable, and statistics almost impossible to estimate from finite data. Convolutional neural networks (CNNs) will play a crucial role in projecting down the dimensionality of our inputs.

For context, we rely on spatial *neighborhoods*. Distance in geographic space provides a form of weak supervision: we assume that tiles that are close together have similar semantics and therefore should, on average, have more similar representations than tiles that are far apart. By exploiting this fact that landscapes in remote sensing datasets are highly spatially correlated, we hope to extract enough learning signal to reliably train deep neural networks.

2.2 Unsupervised triplet loss

To learn a mapping from image tiles to low-dimensional embeddings, we train a convolutional neural network on triplets of tiles, where each triplet consists of an anchor tile t_a, a neighbor tile t_n that is close geographically, and a distant tile t_d that is farther away. Following our distributional assumption, we want to minimize the Euclidean distance between the embeddings of the anchor tile and the neighbor tile, while maximizing the distance between the anchor and distant embeddings. For each tile triplet $\mathbf{t} = (t_a, t_n, t_d)$, we seek to minimize the triplet loss

$$L(\mathbf{t}) = [||f_\theta(t_a) - f_\theta(t_n)||_2 - ||f_\theta(t_a) - f_\theta(t_d)||_2 + m]_+ \quad (1)$$

To prevent the network from pushing the distant tile farther without restriction, we introduce a rectifier $[\cdot]_+$ with margin m. Once the distance to the distant embedding exceeds the distance to the neighbor embedding by at least the margin, we are satisfied. Here, f_θ is a CNN with parameters θ that maps from the domain of image tiles $\mathcal{X}$ to d-dimensional real-valued vector representations, $f_\theta : \mathcal{X} \to \mathbb{R}^d$.

Notice that when $||f_\theta(t_a) - f_\theta(t_n)||_2 < ||f_\theta(t_a) - f_\theta(t_d)||_2$, all embeddings can be scaled by some constant in order to satisfy the margin and bring the loss to zero. We observe this behavior empirically — beyond a small number of iterations, the CNN learns to increase embedding magnitudes and the loss decreases to zero. By penalizing the embeddings' l^2-norms, we constrain the network to generate embeddings within a hypersphere and encourage a better representation, not just a bigger one. Given a dataset of N tile triplets, our full training objective is

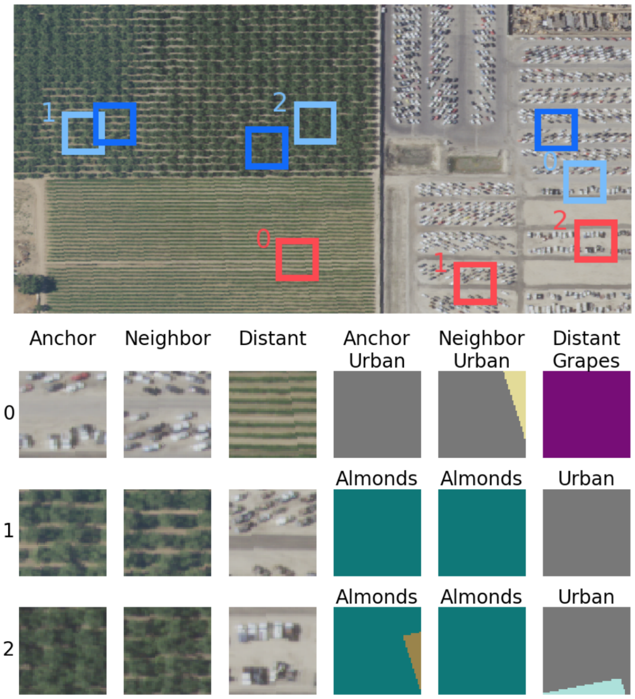

Figure 1: **Top:** Light blue boxes denote anchor tiles, dark blue neighbor tiles, and red distant tiles. **Bottom:** Tile triplets corresponding to the top panel. The columns show anchor, neighbor, and distant tiles and their respective CDL class labels. Anchor and neighbor tiles tend to be the same class, while anchor and distant tend to be different.

$$\min_\theta \sum_{i=1}^{N} \left[L(\mathbf{t}^{(i)}) + \lambda \left(||z_a^{(i)}||_2 + ||z_n^{(i)}||_2 + ||z_d^{(i)}||_2 \right) \right], \quad (2)$$

where λ controls the regularization strength and $z_a^{(i)} = f_\theta(t_a^{(i)}) \in \mathbb{R}^d$ and similarly for $z_n^{(i)}$ and $z_d^{(i)}$.

2.3 Triplet sampling

The sampling procedure for t_a, t_n, and t_d is described by two parameters:

- **Tile size** defines the pixel width and height of a single tile.
- **Neighborhood** defines the region around the anchor tile from which to sample the neighbor tile. In our implementation, if the neighborhood is 100 pixels, then the center of the neighbor tile must be within 100 pixels of the anchor tile center both vertically and horizontally. The distant tile is sampled at random from outside this region.

Tile size should be chosen so that tiles are large enough to contain information at the scale needed for downstream tasks. Neighborhood should be small enough that neighbor tiles will be semantically similar to the anchor tile, but large

Algorithm 1 SampleTileTriplets(D, N, s, r)

```
Input: Image dataset $D$, number of triplets $N$, tile size $s$, neighborhood radius $r$
Output: Tile triplets $T = \{(t_a^{(i)}, t_n^{(i)}, t_d^{(i)})\}_{i=1}^N$

Initialize tile triplets $T = \{\}$
for $i \leftarrow 1, N$ do
    $t_a^{(i)} \leftarrow$ SampleTile$(D, s)$
    $t_n^{(i)} \leftarrow$ SampleTile(Neighborhood$(D, r, t_a^{(i)}), s)$
    $t_d^{(i)} \leftarrow$ SampleTile($\neg$Neighborhood$(D, r, t_a^{(i)}), s)$
    Update $T \leftarrow T \cup (t_a^{(i)}, t_n^{(i)}, t_d^{(i)})$
end for
return $T$

function SampleTile$(A, s)$
    $t \leftarrow$ Sample tile of size $s$ uniformly at random from $A$
    return $t$
end function

function Neighborhood$(D, r, t)$
    $A \leftarrow$ Subset of $D$ within radius $r$ of tile $t$
    return $A$
end function
```

enough to capture intra-class (and potentially some inter-class) variability. In practice, we find that plotting some example triplets as in Fig. 1 allowed us to find reasonable values for these parameters. Results across tile size and neighborhood on our land cover classification experiment are reported in Table A3.[1]

Pseudocode for sampling a dataset of triplets is given in Algorithm 1. Note that no knowledge of actual geographical locations is needed, so Tile2Vec can be applied to any dataset without knowledge of the data collection procedure.

2.4 Scalability

Like most deep learning algorithms, the Tile2Vec objective (Eq. 2) allows for mini-batch training on large datasets. More importantly, the use of the triplet loss allows the training dataset to grow with a *quadratic* relationship relative to the size of the available remote sensing data. Concretely, assume that for a given remote sensing dataset we have a sampling budget of N triplets. If we train using the straightforward approach of Eq. 2, we will iterate over N training examples in each epoch. However, we notice that in most cases the area covered by our dataset is much larger than the area of a single neighborhood. For any tile t, the likelihood that any particular t' in the other $(N-1)$ tiles is in its neighborhood is extremely low. Therefore, at training time we can match any (t_a, t_n) pair with any of the $3N$ tiles in the dataset to increase the number of unique example triplets that the network sees from $\mathcal{O}(N)$ to $\mathcal{O}(N^2)$.

In practice, we find that combining Tile2Vec with this data augmentation scheme to create massive datasets results in an algorithm that is easy to train, robust to hyperparameter choices, and resistant to overfitting. This point will be revisited in section 4.1.

3 Datasets

We evaluate Tile2Vec on several widely-used classes of remote sensing imagery, as well as a non-image dataset of country characteristics. A brief overview of data organized by experiment is given here, with more detailed descriptions in Appendix A.5.

3.1 Land cover classification

We first evaluate Tile2Vec on a land use classification task — predicting what is on the Earth's surface from remotely sensed imagery — that uses the following two datasets: The USDA's **National Agriculture Imagery Program (NAIP)** provides aerial imagery for public use that has four spectral bands — red (R), green (G), blue (B), and infrared (N) — at 0.6 m ground resolution. We obtain an image of Central Valley, California near the city of Fresno for the year 2016 (Fig. 2), spanning latitudes $[36.45, 37.05]$ and longitudes $[-120.25, -119.65]$. The **Cropland Data Layer (CDL)** is a raster geo-referenced land cover map collected by the USDA for the continental United States (USDA-NASS 2016). Offered at 30 m resolution, it includes 132 class labels spanning crops, developed areas, forest, water, and more. In our NAIP dataset, we observe 66 CDL classes (Fig. A10). We use CDL as ground truth for evaluation by upsampling it to NAIP resolution.

3.2 Latent space interpolation and visual analogy

We explore Tile2Vec embeddings by visualizing linearly interpolated tiles in the learned feature space and performing visual analogies on two datasets. Tiles sampled from **NAIP** are used in the latent space interpolation evaluation. The USGS and NASA's **Landsat 8 satellite** provide moderate-resolution (30 m) multispectral imagery on a 16-day collection cycle. Landsat datasets are public and widely used in agricultural, environmental, and other scientific applications. We generate median Landsat 8 composites containing 7 spectral bands over the urban and rural areas of three major US cities — San Francisco, New York City, and Boston — for a visual analogy evaluation.

3.3 Poverty prediction in Uganda

We evaluate the regression task of predicting local poverty levels from **Landsat 7** composites of Uganda from 2009-2011 containing 5 spectral bands. The World Bank's **Living Standards Measurement Study (LSMS)** surveys measure annual consumption expenditure at the household and village levels — these measurements are the basis for determining international standards for extreme poverty. We use the Uganda 2011-12 survey as labels for the poverty prediction task described in (Jean et al. 2016).

3.4 Worldwide country health index prediction

Lastly, to demonstrate that Tile2Vec can be used with other high-dimensional vector data within a spatial context, we predict a subset of country characteristics from other country features. The **CIA World Factbook** is an annual document

[1] Appendix available at https://arxiv.org/abs/1805.02855.

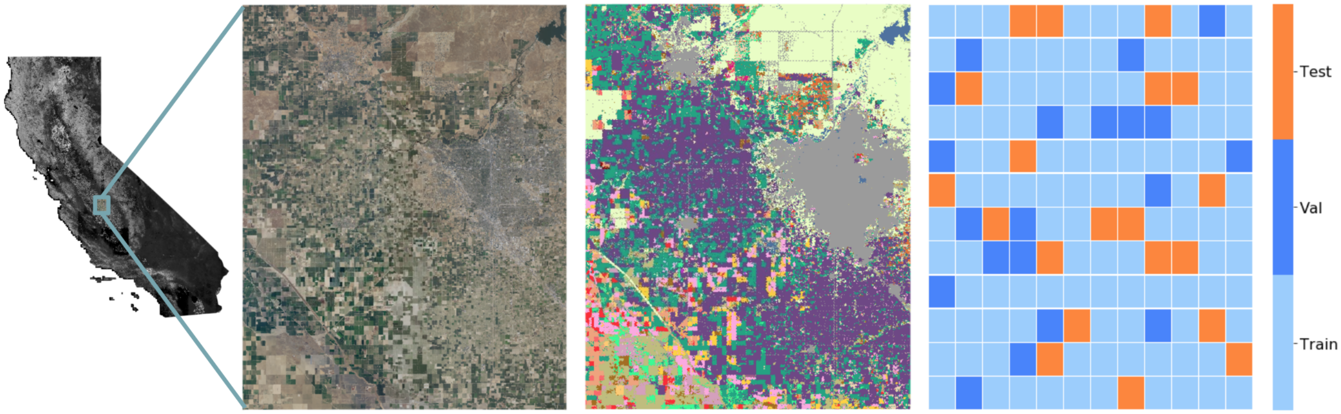

Figure 2: **Left:** Our NAIP aerial imagery covers 2500 km^2 around Fresno, California. **Center:** Land cover types as labeled by the Cropland Data Layer (CDL, see Section 3.1) show a highly heterogeneous landscape; each color represents a different CDL class. **Right:** For the land cover classification task, we split the dataset spatially into train, validation, and test sets.

compiled by the U.S. Central Intelligence Agency containing information on the governments, economies, energy systems, and societies of 267 world entities (Factbook 2015). We extract a dataset from the 2015 Factbook that contains 73 real-valued features (e.g., infant mortality rate, GDP per capita, crude oil production) for 242 countries.

4 Experiments

4.1 Land cover classification using aerial imagery

We train Tile2Vec embeddings on 100k triplets sampled from the NAIP dataset. The Tile2Vec CNN is a ResNet-18 architecture (He et al. 2016) modified for 28 × 28 CIFAR-10 images (1) with an additional residual block to handle our larger input and (2) without the final classification layer. Each of the 300k 50×50 NAIP tiles is labeled with the mode CDL land cover class and our evaluation metric is classification accuracy on this label.

To ensure that training and test sets are spatially disjoint, we split the area into a 12 × 12 grid of rectangular blocks, which we then partitioned randomly into train (104 blocks), validation (20 blocks), and test (20 blocks) (Fig. 2, right). Each of these blocks is just over 17 km^2 in size, roughly 5000 times the size of each tile. By splitting the dataset at the block level, we are able to reduce the spatial autocorrelation and estimate generalization error with minimal inflation.

Tile2Vec hyperparameter optimization We tune the two main hyperparameters of Algorithm 1 by searching over a grid of tile sizes and neighborhoods. We run the CDL land cover classification experiment 20 times in total, using combinations of tile size in $[25, 50, 75, 100]$ and neighborhood radius in $[50, 100, 500, 1000, \text{None}]$, where None indicates that both the neighbor and distant tiles are sampled from anywhere in the dataset (i.e., infinite radius). The resulting accuracies are reported in Table A1. Results suggest that on this task and dataset, a neighborhood radius of 100 pixels strikes the ideal balance between sampling semantically similar tiles and capturing intra-class variability, though classification accuracy remains higher than the model with infinite radius even when the neighborhood is increased to 1000 pixels. Accuracy also increases with tile size, which can be attributed to greater imbalance of labels at larger tile sizes (Appendix A.4) as well as greater available spatial context for classification.

Because CDL labels are at a resolution (30 m) equivalent to 50 NAIP pixels (0.6 m), we ultimately choose a tile size of 50 and neighborhood of 100 pixels for the land cover classification task. For consistency, subsequent experiments also use these default hyperparameters. Although these default hyperparameters yield high performance in most cases, they should generally be optimized for new datasets and tasks.

Unsupervised learning baselines We compare Tile2Vec to a number of unsupervised feature extraction methods. We describe each baseline here, and provide additional training details in Appendix A.1.

- **Autoencoder**: A convolutional autoencoder is trained on all 300k multispectral tiles, split 90% training and 10% validation. We train until the validation reconstruction error flattens; the encoder is then used to embed tiles into the feature space. The autoencoder achieves good reconstructions on the held-out test set (examples in Appendix A.1).
- **Pre-trained ResNet-18**: A modified ResNet-18 was trained on resized CIFAR-10 images and used as a feature extractor. Since CIFAR-10 only has RGB channels, this approach only allows for use of the RGB bands of NAIP and illustrates the limitations of transferring models from natural images to remote sensing datasets.
- **PCA/ICA**: Each RGBN tile of shape $(50, 50, 4)$ is unraveled into a vector of length 10,000 and then PCA/ICA is used to compute the first 10 principal components for each tile.
- **K-means**: Tiles are clustered in pixel space using k-means with $k = 10$, and each tile is represented as 10-dimensional vectors of distances to each cluster centroid.

	$n = 1000$			$n = 10000$		
Unsupervised features	RF	LR	MLP	RF	LR	MLP
Tile2Vec	**52.6 ± 1.1**	**53.7 ± 1.3**	**55.1 ± 1.2**	**56.9 ± 0.3**	**59.7 ± 0.3**	**58.4 ± 0.3**
Autoencoder	49.1 ± 0.7	44.7 ± 1.0	52.0 ± 1.0	53.1 ± 0.2	55.6 ± 0.2	57.2 ± 0.4
Pre-trained ResNet-18	47.7 ± 0.6	48.4 ± 0.8	49.9 ± 1.7	50.6 ± 0.2	53.7 ± 0.2	54.4 ± 0.4
PCA	46.9 ± 0.8	50.2 ± 0.4	43.6 ± 5.3	50.1 ± 0.3	51.1 ± 0.1	52.4 ± 0.3
ICA	47.7 ± 0.6	50.1 ± 0.6	46.7 ± 3.1	50.4 ± 0.4	51.1 ± 0.1	52.5 ± 0.2
K-Means	43.1 ± 0.8	49.4 ± 0.4	44.5 ± 3.9	45.6 ± 0.5	50.0 ± 0.1	50.5 ± 0.2

Table 1: Comparison of Tile2Vec features to unsupervised baselines on the CDL classification task in Section 4.1. Random forest (RF), logistic regression (LR), and multilayer perceptron (MLP) classifiers are trained over 10 trials of $n = 1000$ and $n = 10000$ randomly sampled labels, with mean accuracies and standard deviations reported.

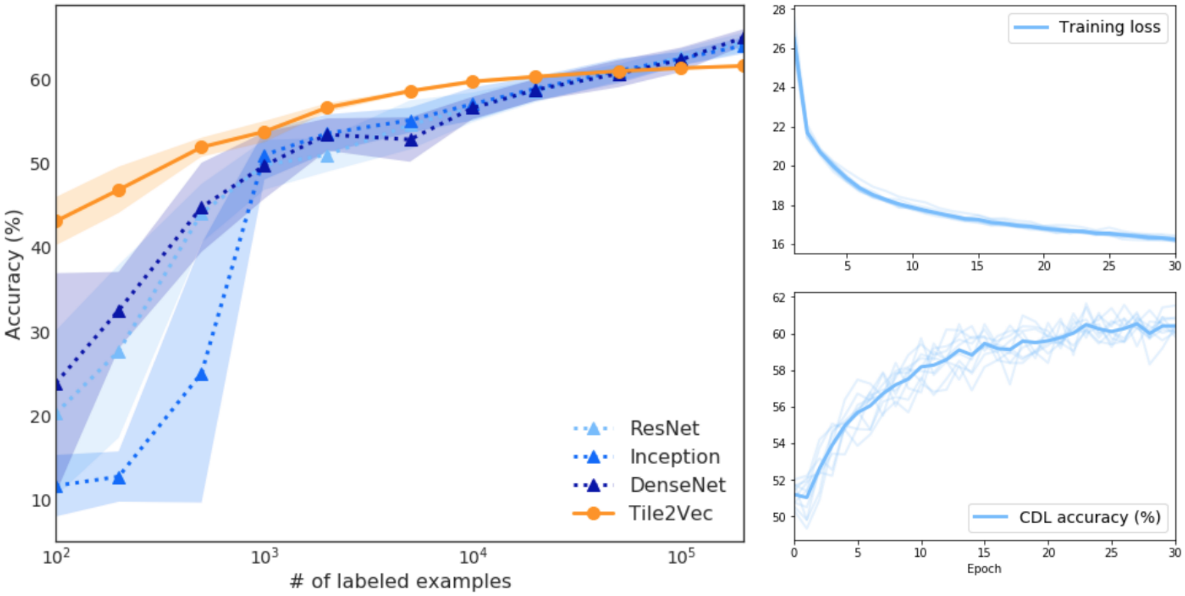

Figure 3: **Left:** Logistic regression on Tile2Vec unsupervised features outperforms supervised CNNs until 50k labeled examples. **Right:** The Tile2Vec triplet loss decreases steadily and downstream classification accuracy tracks the loss.

As shown in Table 1, the features learned by Tile2Vec outperform other unsupervised features when used by random forest (RF), logistic regression (LR), and multilayer perceptron (MLP) classifiers trained on $n = 1000$ or $n = 10000$ labels. We also trained a DCGAN (Radford, Metz, and Chintala 2015) as a generative modeling approach to unsupervised feature learning. Although we were able to generate reasonable samples, features learned by the discriminator performed poorly — samples and results can be found in Appendix A.1. Approaches based on variational autoencoders (VAEs) would also provide intriguing baselines, but we are unaware of existing models capable of capturing complex multispectral image distributions.

Supervised learning comparisons Surprisingly, our Tile2Vec features are also able to outperform fully-supervised CNNs trained directly on the classification task with large amounts of labeled data. Fig. 3 shows that applying logistic regression on Tile2Vec features beats several state-of-the-art supervised architectures (He et al. 2016; Szegedy et al. 2015; Huang et al. 2017) trained on as many as 50k CDL labels. We emphasize that the Tile2Vec CNN and the supervised ResNet share the same architecture, so logistic regression in Fig. 3 is directly comparable to the classification layers of the supervised architectures. Similar results for random forest and multilayer perceptron classifiers can be found in Appendix A.4.

Latent space interpolation We further explore the learned representations with a latent space interpolation experiment shown in Fig. 4. Here, we start with the Tile2Vec embeddings of a field tile and an urban tile and linearly interpolate between the two. At each point along the interpolation, we search for the five nearest neighbors in the latent space and display the corresponding tiles. As we move through the semantically meaningful latent space, we recover tiles that are more and more developed.

Training details Tile2Vec is easy to train and robust to the choice of hyperparameters. We experimented with margins

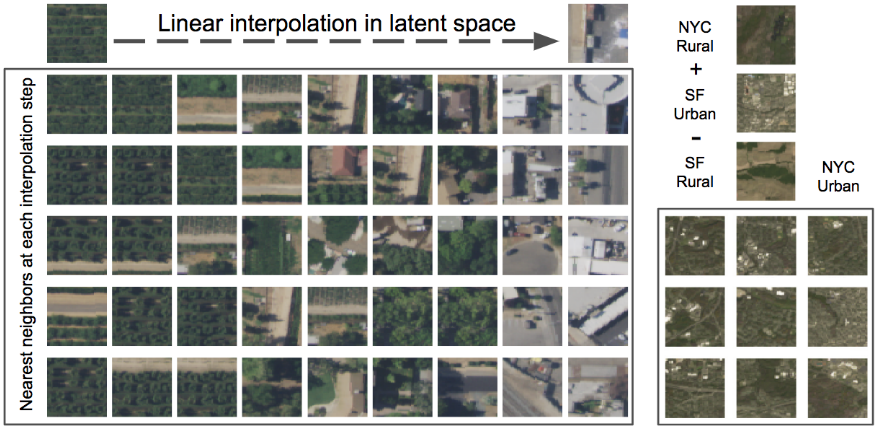
Linear interpolation in latent space
Nearest neighbors at each interpolation step
NYC
Rural
+
SF
Urban
-
SF
Rural
NYC
Urban

Figure 4: **Left:** Linear interpolation in the latent space at equal intervals between representations of rural and urban images. Below, we show 5 nearest neighbors in the latent space to each interpolated vector. **Right:** Starting with a rural NYC embedding, we add urban SF and subtract rural SF to successfully discover urban NYC tiles. More visual analogies are shown in Fig. A6.

ranging from 0.1 to 100 and found little effect on accuracy. Using a margin of 50, we trained Tile2Vec for 10 trials with different random initializations and show the results in Fig. 3. The training loss is stable from epoch to epoch, consistently decreasing, and most importantly, a good proxy for unsupervised feature quality as measured by performance on the downstream task (Fig. 3, bottom right). By combining explicit regularization (Eq. 2) with the data augmentation scheme described in Section 2.4, we observe that Tile2Vec does not seem to overfit even when trained for many epochs.

4.2 Visual analogies across US cities

To evaluate Tile2Vec qualitatively, we explore three major metropolitan areas of the United States: San Francisco, New York City, and Boston. First, we train a Tile2Vec model on the San Francisco dataset only. Then we use the trained model to embed tiles from all three cities. As shown in Fig. 4 and A6, these learned representations allow us to perform arithmetic in the latent space, or *visual analogies* (Radford, Metz, and Chintala 2015). By adding and subtracting vectors in the latent space, we can recover image tiles that are semantically expected given the operations applied.

Here we use Landsat images with 7 spectral bands, demonstrating that Tile2Vec can be applied effectively to highly multispectral datasets. Tile2Vec can also learn representations at multiple scales: each Landsat 8 (30 m resolution) tile covers 2.25 km^2, while the NAIP and DigitalGlobe tiles are 2500 times smaller. Finally, Tile2Vec learns robust representations that allow for domain adaptation or transfer learning, as the three datasets have widely varying spectral distributions (Fig. A9).

4.3 Poverty prediction from satellite imagery

Next, we apply Tile2Vec to predict annual consumption expenditures in Uganda from satellite imagery. Accurate measurements of poverty are essential for both research and policy, but reliable data is limited in the developing world — machine learning methods that are still effective when labeled data is scarce could help to fill these critical gaps.

Features	d	kNN	RF	RR
Tile2Vec	10	**77.5** $\pm$ **1.0**	**76.0** $\pm$ **1.3**	**69.6** $\pm$ **1.0**
Non-health	60	62.8 $\pm$ 1.5	72.1 $\pm$ 1.6	68.7 $\pm$ 1.7
Locations	2	69.3 $\pm$ 1.0	67.7 $\pm$ 2.6	11.6 $\pm$ 1.5

Table 2: Predicting health index using Tile2Vec features versus non-health features and locations (i.e., $\{$lat, lon$\}$). Here, d is feature dimension, kNN is k-nearest neighbors, RF is random forest, and RR is ridge regression. Hyperparameters (e.g., k and regularization strength) are tuned for each feature set. We report average r^2 and standard deviation for 10 trials of 3-fold cross-validation.

The previous state-of-the-art result used a transfer learning approach in which a CNN is trained to predict nighttime lights (a proxy for poverty) from daytime satellite images — the features from this model are then used to predict consumption expenditures (Jean et al. 2016). We use the same LSMS survey preprocessing pipeline and ridge regression evaluation (see Appendix A.2). Evaluating over 10 trials of 5-fold cross-validation, we report an average r^2 of 0.496 ± 0.014 compared to $r^2 = 0.41$ for the transfer learning approach — this is achieved with publicly available imagery with much lower resolution than the proprietary images used in (Jean et al. 2016) (30 m vs. 2.4 m).

4.4 Generalizing to other spatial data: Predicting country health indices

The CIA Factbook contains 73 features spanning economic, energy, social, and other characteristics of countries around the world. To demonstrate that Tile2Vec can leverage spatial coherence for non-image datasets as well, we use 13 of the features in the CIA Factbook related to public health and compute a health index, then attempt to predict this health index from the remaining 60 features. We train Tile2Vec by sampling triplets of countries and feeding the feature vectors into a small MLP with one hidden layer.

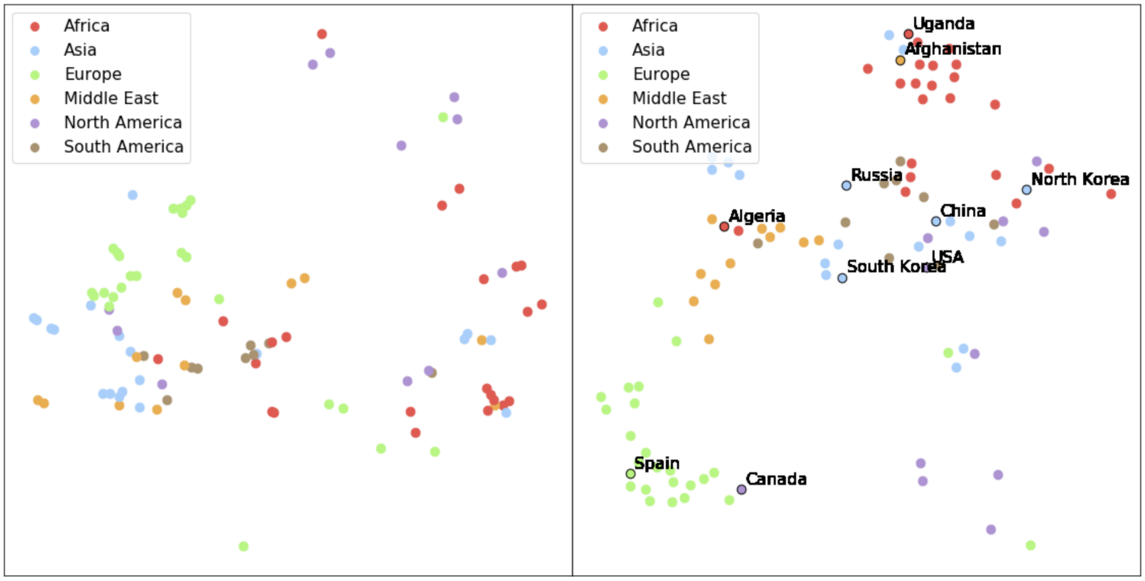

Figure 5: **Left:** The 60 non-health country features visualized using t-SNE. Spatial relationships are preserved for some clusters, but not for others. **Right:** The 10-dimensional Tile2Vec embeddings visualized using t-SNE. The latent space now respects both spatial and characteristic similarities. Several countries are annotated to highlight interesting relationships: North Korea and South Korea are embedded far apart even though they are spatial neighbors; USA, South Korea, and China are embedded close together though they are geographically separated.

As shown in Table 2, the embeddings learned by Tile2Vec on this small spatial dataset ($N = 242$) outperform both the original features and approaches that explicitly use spatial information. Fig. 5 shows the original 60-dimensional feature vectors as well as the 10-dimensional learned Tile2Vec embeddings projected down to two dimensions using t-SNE (van der Maaten and Hinton 2008). While there is some geographic grouping of countries in projecting down the original features, the Tile2Vec embeddings appear to capture both geographic proximity and socioeconomic similarity.

In this experiment, the Haversine formula was used to compute the great-circle distance between pairs of countries in kilometers — future work could explore using distance functions more meaningful to the application at hand, e.g., shared borders or trade volume.

5 Related Work

Our inspiration for using spatial context to learn representations originated from continuous word representations like Word2vec and GloVe (Mikolov et al. 2013b; 2013a; Pennington, Socher, and Manning 2014). In NLP, the distributional hypothesis can be summarized as "a word is characterized by the company it keeps" — words that appear in the same context likely have similar semantics. We apply this concept to remote sensing data, with multispectral image tiles as the atomic unit analogous to individual words in NLP, and geospatial neighborhoods as the "company" that these tiles keep. A related, supervised version of this idea is the patch2vec algorithm (Fried, Avidan, and Cohen-Or 2017), which its authors describe as learning "globally consistent image patch representations". Working with natural images, they use a very similar triplet loss (first introduced in (Hoffer and Ailon 2015)), but sample their patches with supervision from an annotated semantic segmentation dataset.

Unsupervised learning for visual data is an active area of research and thus impossible to summarize concisely, but we attempt a brief overview of the most relevant topics here. The three main classes of deep generative models — likelihood-based variational autoencoders (VAEs) (Kingma and Welling 2013), likelihood-free generative adversarial networks (GANs) (Goodfellow et al. 2014), and various autoregressive models (Oord, Kalchbrenner, and Kavukcuoglu 2016; van den Oord et al. 2016) — attempt to learn the generating data distribution from training samples. Other related lines of work use spatial or temporal context to learn high-level image representations. Some strategies for using spatial context involve predicting the relative positions of patches sampled from within an image (Noroozi and Favaro 2016; Doersch, Gupta, and Efros 2015) or trying to fill in missing portions of an image (in-painting) (Pathak et al. 2016). In videos, nearby frames can be used to learn temporal embeddings (Ramanathan et al. 2015); other methods leveraging the temporal coherence and invariances of videos for feature learning have also been proposed (Misra, Zitnick, and Hebert 2016; Wang and Gupta 2015).

6 Conclusion

We demonstrate the efficacy of Tile2Vec as an unsupervised feature learning algorithm for spatially distributed data on tasks from land cover classification to poverty prediction. Our method can be applied to image datasets spanning moderate to high resolution, RGB or multispectral bands, and collected via aerial or satellite sensors, and even to non-image datasets. Tile2Vec outperforms other unsupervised feature extraction techniques on a difficult classification task — surprisingly, it even outperforms supervised CNNs trained on 50k labeled examples.

In this paper, we focus on exploiting *spatial* coherence, but many geospatial datasets also include sequences of data collected over time. Temporal patterns can be highly informative (e.g., seasonality, crop cycles), and we plan to explore this aspect in future work. Remote sensing data have largely been unexplored by the machine learning community — more research in these areas could result in enormous progress on many problems of global significance.

Acknowledgements

This research was supported by NSF (#1651565, #1522054, #1733686), ONR, Sony, and FLI. NJ was supported by the Department of Defense (DoD) through the National Defense Science & Engineering Graduate Fellowship (NDSEG) Program.

References

Doersch, C.; Gupta, A.; and Efros, A. A. 2015. Unsupervised visual representation learning by context prediction. In *Proceedings of the IEEE International Conference on Computer Vision*, 1422–1430.

Factbook, C. 2015. The World Factbook; 2010. http://www.cia.gov/library/publications/the-world-factbook. Accessed: 2018-01-20.

Foody, G. M. 2003. Remote sensing of tropical forest environments: Towards the monitoring of environmental resources for sustainable development. *International Journal of Remote Sensing* 24(20):4035–4046.

Fried, O.; Avidan, S.; and Cohen-Or, D. 2017. Patch2Vec: Globally consistent image patch representation. In *Computer Graphics Forum*, volume 36, 183–194. Wiley Online Library.

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, 2672–2680.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 770–778.

Hoffer, E., and Ailon, N. 2015. Deep metric learning using triplet network. In *International Workshop on Similarity-Based Pattern Recognition*, 84–92. Springer.

Huang, G.; Liu, Z.; Weinberger, K. Q.; and van der Maaten, L. 2017. Densely connected convolutional networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, volume 1, 3.

Jean, N.; Burke, M.; Xie, M.; Davis, W. M.; Lobell, D. B.; and Ermon, S. 2016. Combining satellite imagery and machine learning to predict poverty. *Science* 353(6301):790–794.

Kingma, D. P., and Welling, M. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Krizhevsky, A.; Sutskever, I.; and Hinton, G. 2012. ImageNet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, 1097–1105.

Levy, O., and Goldberg, Y. 2014. Neural word embedding as implicit matrix factorization. In *Advances in Neural Information Processing Systems*, 2177–2185.

Mikolov, T.; Chen, K.; Corrado, G.; and Dean, J. 2013a. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G. S.; and Dean, J. 2013b. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, 3111–3119.

Misra, I.; Zitnick, C. L.; and Hebert, M. 2016. Shuffle and learn: unsupervised learning using temporal order verification. In *European Conference on Computer Vision*, 527–544. Springer.

Mulla, D. J. 2013. Twenty five years of remote sensing in precision agriculture: Key advances and remaining knowledge gaps. *Biosystems Engineering* 114(4):358 – 371. Special Issue: Sensing Technologies for Sustainable Agriculture.

Noroozi, M., and Favaro, P. 2016. Unsupervised learning of visual representations by solving jigsaw puzzles. In *European Conference on Computer Vision*, 69–84. Springer.

Oord, A. v. d.; Kalchbrenner, N.; and Kavukcuoglu, K. 2016. Pixel recurrent neural networks. *arXiv preprint arXiv:1601.06759*.

Oshri, B.; Hu, A.; Adelson, P.; Chen, X.; Dupas, P.; Weinstein, J.; Burke, M.; Lobell, D.; and Ermon, S. 2018. Infrastructure quality assessment in africa using satellite imagery and deep learning. *Proc. 24th ACM SIGKDD Conference*.

Pathak, D.; Krahenbuhl, P.; Donahue, J.; Darrell, T.; and Efros, A. A. 2016. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2536–2544.

Pennington, J.; Socher, R.; and Manning, C. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1532–1543.

Radford, A.; Metz, L.; and Chintala, S. 2015. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*.

Ramanathan, V.; Tang, K.; Mori, G.; and Fei-Fei, L. 2015. Learning temporal embeddings for complex video analysis. In *Proceedings of the IEEE International Conference on Computer Vision*, 4471–4479.

Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A.; et al. 2015. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

USDA-NASS. 2016. USDA National Agricultural Statistics Service Cropland Data Layer. published crop-specific data layer [online].

van den Oord, A.; Kalchbrenner, N.; Espeholt, L.; Vinyals, O.; Graves, A.; et al. 2016. Conditional image generation with pixelcnn decoders. In *Advances in Neural Information Processing Systems*, 4790–4798.

van der Maaten, L., and Hinton, G. 2008. Visualizing high-dimensional data using t-SNE. *Journal of Machine Learning Research* 9:2579–2605.

Wang, X., and Gupta, A. 2015. Unsupervised learning of visual representations using videos. *arXiv preprint arXiv:1505.00687*.

Xie, M.; Jean, N.; Burke, M.; Lobell, D.; and Ermon, S. 2016. Transfer learning from deep features for remote sensing and poverty mapping. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, AAAI'16, 3929–3935. AAAI Press.