Temporal Cluster Matching for Change Detection of Structures from Satellite Imagery

Caleb Robinson
caleb.robinson@microsoft.com
Microsoft AI for Good Research Lab
USA

Anthony Ortiz
anthony.ortiz@microsoft.com
Microsoft AI for Good Research Lab
USA

Juan M. Lavista Ferres
jlavista@microsoft.com
Microsoft AI for Good Research Lab
USA

Brandon Anderson
branders@law.stanford.edu
Stanford RegLab
USA

Daniel E. Ho
dho@law.stanford.edu
Stanford RegLab
USA

ABSTRACT

Longitudinal studies are vital to understanding dynamic changes of the planet, but labels (e.g., buildings, facilities, roads) are often available only for a single point in time. We propose a general model, Temporal Cluster Matching (TCM), for detecting *building changes* in time series of remotely sensed imagery when footprint labels are observed only once. The intuition behind the model is that the relationship between spectral values inside and outside of building's footprint will change when a building is constructed (or demolished). For instance, in rural settings, the pre-construction area may look similar to the surrounding environment until the building is constructed. Similarly, in urban settings, the pre-construction areas will look different from the surrounding environment until construction. We further propose a heuristic method for selecting the parameters of our model which allows it to be applied in novel settings without requiring data labeling efforts (to fit the parameters). We apply our model over a dataset of poultry barns from 2016/2017 high-resolution aerial imagery in the Delmarva Peninsula and a dataset of solar farms from a 2020 mosaic of Sentinel 2 imagery in India. Our results show that our model performs as well when fit using the proposed heuristic as it does when fit with labeled data, and further, that supervised versions of our model perform the best among all the baselines we test against. Finally, we show that our proposed approach can act as an effective data augmentation strategy – it enables researchers to augment existing structure footprint labels along the time dimension and thus use imagery from multiple points in time to train deep learning models. We show that this improves the spatial generalization of such models when evaluated on the same change detection task.

CCS CONCEPTS

• **Computing methodologies** → **Machine learning**; **Unsupervised learning**; **Computer vision representations**.

COMPASS '21, June 28-July 2, 2021, Virtual Event, Australia

ACM ISBN 978-1-4503-8453-7/21/06…$15.00
https://doi.org/10.1145/3460112.3471952

KEYWORDS

change detection, building footprints, deep learning, clustering

ACM Reference Format:
Caleb Robinson, Anthony Ortiz, Juan M. Lavista Ferres, Brandon Anderson, and Daniel E. Ho. 2021. Temporal Cluster Matching for Change Detection of Structures from Satellite Imagery. In *ACM SIGCAS Conference on Computing and Sustainable Societies (COMPASS) (COMPASS '21), June 28-July 2, 2021, Virtual Event, Australia.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3460112.3471952

1 INTRODUCTION

Catalogs of high-resolution remotely sensed imagery have become increasingly available to the scientific community. The availability of such imagery has revolutionized scientific fields and society at large. For example, 1m resolution aerial imagery from the US Department of Agriculture (NAIP imagery) has been released on a 2-year rolling basis over the entire US for over a decade and the commercial satellite imagery provider, Planet, recently started to release 5m satellite imagery covering the whole tropical forest region of the world on a monthly basis. One estimate is that the opening of Landsat imagery in 2008 led to the creation of $3.45B in economic value in 2017 alone [27]. The accumulation of such data facilitates an entirely new branch of longitudinal studies – analyzing the Earth and how it has changed over time.

As the climate, technology, and human population change on an ever more rapid timescale, such longitudinal studies become particularly vital to understanding the past, present, and future of the environment. Despite the usefulness of time series data, such research faces two important practical challenges. First, the large labeled datasets that have fueled advances in computer vision are much more limited in the satellite imagery context [5, 6, 25]. Second, efforts in creating labeled data from remotely sensed imagery are typically focused on a single point in time [6, 25, 36]. Project requirements may only call for a single layer of labels, or budget constraints may limit the number of labels that can be generated. This has the effect of creating labeled datasets that are "frozen" in time. Expanding such "frozen" datasets to multiple points in time in independent follow-up work can be resource-intensive and difficult, as the same image-preprocessing and labeling methodology steps used in the original work need to be precisely reproduced in order to generate comparable data.

Going beyond "frozen" datasets would enable a wide range of *temporal* inferences from satellite imagery, with significant social, economic, and policy implications. Previous studies include the detection of urban expansion [35], zoning violations [23], habitat modification [7], compliance with agricultural subsidies [20], construction on wetlands [12], and damage assessments from natural disasters [9, 10, 19].

Algorithmic approaches for expanding "frozen" datasets can thus be useful in facilitating ecological and policy-based analysis. In this work we propose a simple model, Temporal Cluster Matching (TCM), for determining *when* structures were previously constructed given a labeled dataset of structure footprints generated from imagery captured at a particular point in time. This model, importantly, does not rely on the differences in spectral values between layers of remotely sensed imagery as there can be considerable variance in these values depending on imaging conditions, the type of sensor used, etc. Instead, it compares a representation of the spectral values inside a building footprint to a representation of the spectral values in the surrounding area for each point in the time series. Whenever the distribution of spectral values within the footprint becomes dissimilar to that of its surroundings then the footprint is likely to have been developed. We further propose a method for fitting the parameters of this model which does not rely on additional labeled footprint data over time and show that this "semi-supervised TCM" performs comparably to supervised methods.

Specifically, we demonstrate the performance of this algorithm in two distinct settings:

(1) Poultry barns from concentrated animal feeding operations (CAFOs) in the United States, using high-resolution aerial imagery from the National Agricultural Imagery Program (NAIP), and
(2) Solar farm footprints in the Indian state of Karnataka, using Sentinel 2 annual mosaics.

Both settings are of significant environmental importance and are ripe for longitudinal study. First, CAFOs can have profound effects on water quality and human health in their proximity [22]. Nitrates and other potentially harmful chemicals can, for example, make their way into the groundwater, spreading to adjacent wells and bodies of water over timescales that range into decades. Usage of antibiotics for growth promotion can lead to resistant bacterial infections in nearby populations [2]. Effective regulation in either scenario requires differentiation of these contaminant sources, which, in turn, depends on accurate historical labels and spatio-temporal modeling.

Second, understanding the growth of solar systems is increasingly important in the transition toward clean energy. India is an important example of this, as it has set ambitious goals of generating 450 GW of renewable energy by 2030 with 175 GW deployment by 2022 [8]. Achieving this goal will require an expansion of solar farm installations throughout the country and policy makers will be able to determine better the effects of country-wide efforts with solar farm change data that can be updated year-over-year in a consistent manner. Understanding such solar expansion may also enable more targeted investments for solar potential [21].

To summarize, our contributions include:

- A lightweight model, Temporal Cluster Matching, for detecting when structures are developed in a time-series of remotely sensed imagery, as well as a heuristic method for fitting the parameters of the model. Combined, this results in a proposed approach that only relies on labeled building footprints for a single point in time.
- A series of baseline methods, both supervised and semi-supervised, to evaluate our proposed approach against.
- Experiments comparing our model to the baseline models in two datasets: poultry barn footprints with aerial imagery, and solar farm footprints with satellite imagery.
- A code release that allows users to run the model in novel settings, as well as scripts for reproducing our experiments: https://github.com/microsoft/temporal-cluster-matching

2 RELATED WORK

Our work pertains to several different literatures. First, much work has focused on methods for detection of building footprints. For instance, [39] uses Mask R-CNN to train a model to detect buildings while [37] uses a semantic segmentation model (U-Net [24]) to segment buildings in imagery. While deep learning approaches have made rapid advances, they have largely been focused on static inferences. Moving to a different domain (spatially or temporally) can prove challenging due to shifts in the input distribution (what the input images look like), co-registration errors, and shifts in the target (what buildings looks like) [31]. Zhang et al., for instance, note these as some of the challenges faced when detecting structures at a global scale [38].

Second, other research has focused on detecting changes in satellite imagery. Historically, the remote sensing literature has started from pixel-wise change-point detection – detecting when change happens in a time series of repeated observations of the same location in space. Much work has focused on how to model the characteristics of these time series such as seasonality, changes in illumination, atmospheric conditions, etc [1]. A popular method for performing this task, BFAST, models a time series of remotely sensed observations with trend, seasonal, and remainder components, then uses an unsupervised iterative method to detect change points based on the model [34]. This method relies on observing a relatively long time series, e.g. that shows seasonal components, and thus is not applicable to time series with few data points. Change-point detection can also be performed on other units of analysis. [30] review the literature and organize methods based on their unit of analysis, e.g. pixel-based, object-based, kernel-based, and based on their method for comparing scenes, e.g. based on differencing, transformation, or modeling. Within this organization, our work is the most similar to those that operate on image-object overlays [14, 16, 29] whereby a single segmentation is applied to all imagery in a time series and change at an object level is computed. We use similar ideas in designing our baseline approaches (Section 3.4).

The computer vision and machine learning literature also addresses similar problems. Several methods use a fully supervised approach to detect changes in known building locations: [15] use a supervised approach with decision trees to classify whether a

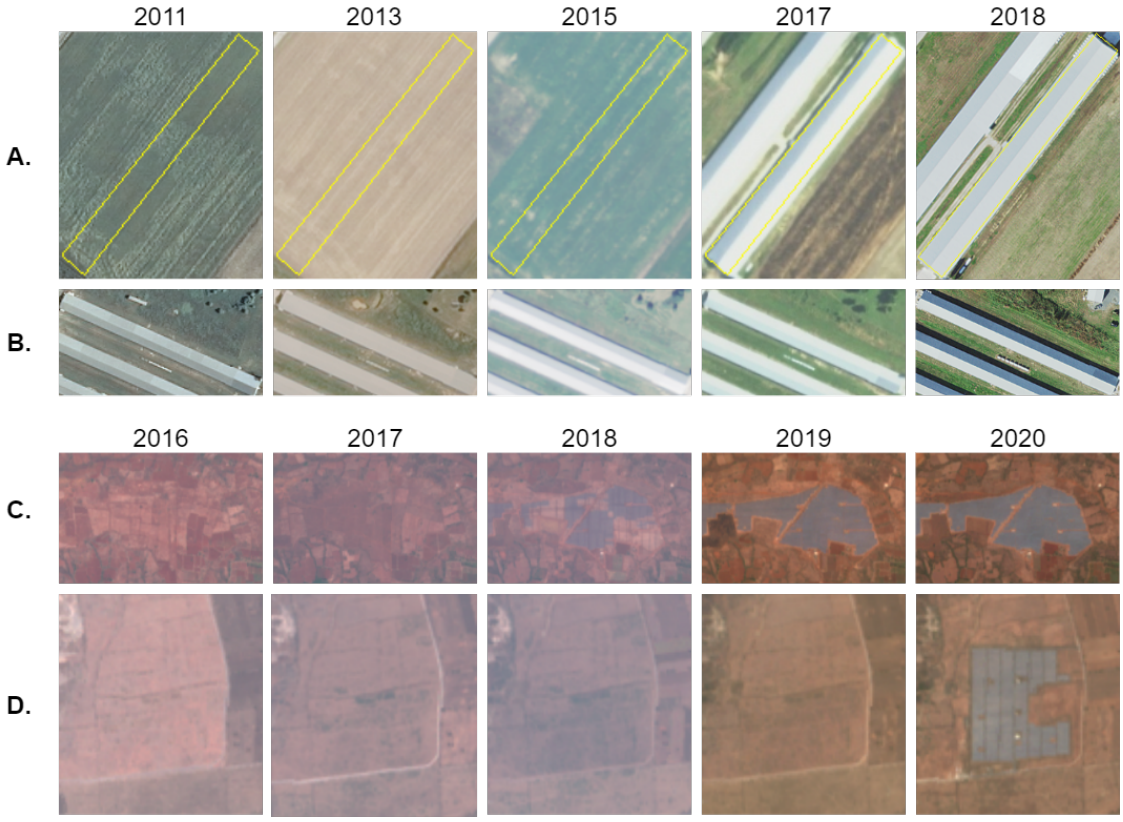

Figure 1: (A and B) Examples of two poultry barn footprints over 5 years of NAIP imagery. We observe inter-year variability of NAIP imagery and the change in the relation of color/texture between the footprint and neighborhood when a footprint is "developed". (C and D) Examples of two solar farm footprints over 5 years of Sentinel 2 imagery. Note, in A we outline the building footprint location in yellow through the entire series of imagery, but omit this outline in remaining rows.

building change occurred from pairs of imagery and [17] use support vector machines to provide estimates for which buildings have changed. Other methods perform a superpixel segmentation step to create objects in pairs of imagery, then model change over these objects with a Markov random field [18]. Most recently, advances in deep learning have driven end-to-end pipelines in building change detection. [3], for instance, uses a Generative Adversarial Network (GAN) to overcome the limitations of pixel-level inferences. These methods all rely on having existing labeled data on either when changes have occurred, or on unchanged areas in consecutive pairs of imagery.

Finally, numerous research teams have provided benchmark datasets for evaluating models at a single point in time [6, 25, 36]. Few datasets provide longitudinal information about the same location over time, so a typical research approach has been to train a model on labeled imagery from one period. [11], for instance, assess the growth of intensive livestock farms in North Carolina, but do so using a model trained on images of such facilities for a single period of time. A notable exception to this is the recent SpaceNet 7 dataset/challenge [33]. This dataset includes 24 multi-spectral images at a 4m/px spatial resolution as well as building footprint labels over time for over 100 unique locations around the world. It is particularly challenging for object based change detection approaches as the median building size is 12.1 pixels [33] and the imagery is not perfectly co-registered. Pixel based segmentation models followed by in-depth post-processing methods achieved the top performance in the competition [32], however it is not yet clear how to adapt such methodology to general change detection tasks.

Our approach contributes to this body of work by providing a semi-supervised approach for detecting changes in structures that easily enables researchers to expand a dataset beyond a single time period, hence enabling domain adaptation by efficient sampling of images across time. The approach we propose can be seen as a lightweight, data-driven method to expand "frozen" imagery longitudinally, enabling researchers to address a rich set of dynamic questions.

3 METHODS

3.1 Problem statement

Formally, we would like to find when a structure was developed (or, more generally, changed) given a time series of remotely sensed imagery of it and the surrounding area, $\left[\mathbf{X}^1, \ldots, \mathbf{X}^t\right]$ and its *footprint*, P, at time t. We represent this footprint as a mask, $\mathbf{Y}^t$. Here,

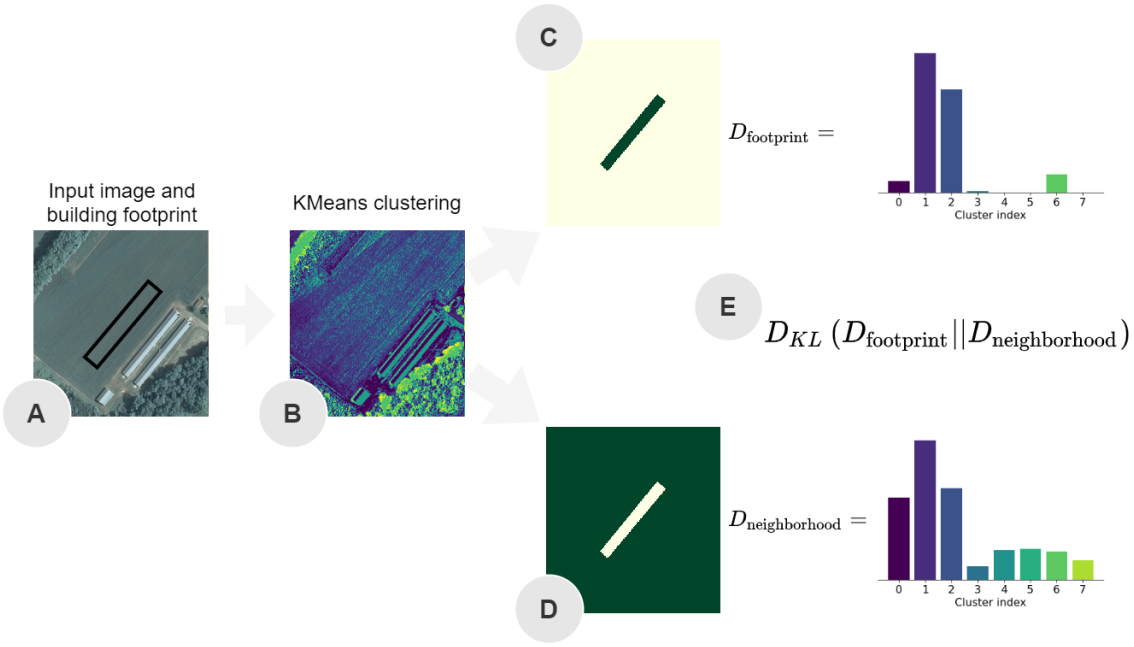

Figure 2: Schematic description of Temporal Cluster Matching: (A) the input imagery and query building footprint; (B) k-means clustering of the input imagery (C, D) discrete distributions of clusters created from the pixels within the footprint polygon and from the pixels outside of the footprint (i.e. the neighborhood); (E) the KL-divergence between the two distributions indicates the similarity of the footprint to its neighborhood.

the i^{th} image in the time series $\mathbf{X}^i \in \mathbb{Z}^{w \times h \times c}$ is a georeferenced image with width, w, height, h, and number of spectral bands, c. Similarly, $\mathbf{Y}^t \in \{0,1\}^{w \times h}$ is a georeferenced binary mask with the same dimensions that contains a 1 in every spatial location that the given structure covers at time t and a 0 elsewhere. We want to estimate the point in time that the structure was created, i.e. find $l \in [1, t]$, where $\mathbf{X}^l$ contains the structure, for the smallest such l. Note that we assume the structure to exist at t.

3.2 Temporal Cluster Matching

Our proposed model, Temporal Cluster Matching, relies on the assumption that when a structure is built its footprint will have a different set of colors and textures than its immediate surroundings compared to when the structure was not built. For example, an undeveloped piece of land in a rural setting will likely contain some sort of vegetation, and that vegetation will probably look similar (in color/texture) to some of its surroundings. When a structure is built on this land, then it will likely look dissimilar to its surroundings (unless e.g. its entire surroundings are also developed at the same time). The same intuition holds in urban environments – an undeveloped piece of land will look dissimilar to its surroundings, however, when it is developed, it will look similar.

We assume that we are given a *footprint*, P, that outlines a structure that has been labeled as developed at time t. Now, we formally define the *neighborhood* of this *footprint*. This *neighborhood* should be larger than the extent of the footprint in order to observe a representative set of colors/textures, so we let r be a radius that serves as a buffer to the building footprint polygon. We then create $\mathbf{Y}^t$ by rasterizing the polygon within this buffered extent and create $\left[\mathbf{X}^0, \ldots, \mathbf{X}^t\right]$ by cropping the same buffered extent from each layer of remotely sensed imagery.

Next, we define a method for comparing the set of colors/textures within the *footprint* to those in the surrounding *neighborhood*. Given a single layer of remotely sensed image from the time series, $\mathbf{X}$, we run k-means to partition the pixels into k clusters. Each pixel can be represented by a set of features that encodes color and texture at its location, for example: the spectral values at the pixel's location, a texture descriptor (such as a local binary pattern) at the location, the spectral values in a window around the location, or some combination of the previous representations. Regardless, the cluster model will assign a cluster index to each pixel in $\mathbf{X}$ which we call $\mathbf{C}$. We then represent an area by the discrete distribution of cluster indices observed in that area. Specifically, we let $D_{\text{footprint}}$ be the distribution of cluster indices from $\mathbf{C}[\mathbf{Y}^t = 1]$ and $D_{\text{neighborhood}}$ be the distribution of cluster indices from $\mathbf{C}[\mathbf{Y}^t = 0]$[1]. Now, we can compare the set of colors/textures within a footprint to those in its surrounding *neighborhood* by calculating the KL-divergence between the two distributions of cluster indices, $d = D_{KL}(D_{\text{footprint}} \,||\, D_{\text{neighborhood}})$. Larger KL-divergence values mean that the color/texture of a footprint is dissimilar to that of its surrounding neighborhood and that it is likely to be developed. We

[1]We use the notation $\mathbf{C}[\mathbf{Y}^t = 1]$ to mean all the cluster indices of pixels where $\mathbf{Y}^t = 1$. We build the discrete distribution by counting the number of pixels assigned to each cluster and normalizing the vector of counts by its sum.

perform this comparison method for each image in the time series to create a list of KL-divergence values $[d_1, \ldots, d_t]$

Finally, we let θ be a threshold value to determine the smallest KL-divergence value that we will consider to indicate a "developed" footprint. More specifically, our model will estimate l as the time that a footprint is first developed for the first l where $d_l > \theta$. This parameter can be found by experimentation using labeled data, or with the heuristic method we describe in Section 3.3. See Algorithm 1 and Figure 2 for an overview of this proposed approach.

Algorithm 1: Temporal Cluster Matching

Input: Time series of remotely sensed imagery, P, k, r, and θ
Output: l, the *first* point in time that the footprint described by P was developed

$[\mathbf{X}^1, \mathbf{X}^1, \ldots, \mathbf{X}^t] \leftarrow$ crop the imagery according to the buffered extent of P by a radius r
$\mathbf{Y}^t \leftarrow$ rasterize P in the same buffered extent
for $l \leftarrow 1$ **to** t **do**
  $\mathbf{C} \leftarrow$ cluster indices from a k-means clustering of $\mathbf{X}^l$ into k clusters
  $D_{\text{footprint}} \leftarrow$ distribution of cluster indices $\mathbf{C}[\mathbf{Y}^t = 1]$
  $D_{\text{neighborhood}} \leftarrow$ distribution of cluster indices $\mathbf{C}[\mathbf{Y}^t = 0]$
  $d \leftarrow D_{KL}(D_{\text{footprint}} \,||\, D_{\text{neighborhood}})$
  if $d > \theta$ **then**
    return l
  end
end
return t

Section 3.4 explores more complex decision models than the single threshold described above, but we note that these require labeled data to fit.

3.3 A heuristic for semi-supervised Temporal Cluster Matching

In application scenarios we would like to use our model, given a dataset of (a) known structure footprints at time t and (b) a time series of remotely sensed imagery over a certain study area, to find when each structure was constructed. Here we propose a method for determining reasonable parameter values for the number of clusters, k, buffer radius, r, and decision threshold, θ, *without assuming that we have prior labeled data on construction dates.*

This heuristic compares the distribution of KL-divergence values calculated by our algorithm for given hyperparameters, k and r, over all footprints at time t (when we assume that structures exist) to the distribution of KL-divergence values over a set of *randomly generated polygons* over the study area. The intuition is that the relationship between random polygons and their *neighborhoods* is similar to the relationship between undeveloped structure footprints and their *neighborhoods*. In other words, this distribution of KL divergence values between color distributions from random polygons and their surroundings will represent what we would expect to observe by chance – i.e. *not* the relationship between the colors in a building footprint and its surroundings. We want to find parameter settings for our algorithm that minimize the overlap between these two distributions because it will make it easier to identify change (see Figure 3 for an illustration of this for poultry barn footprints). Formally, we let p be the distribution of KL-divergence values over footprints at t and q be the distribution of KL-divergence values over random polygons sampled from the study area (over all points in time). These are discrete distributions (e.g after binning KL divergence values) and we can measure the overlap with the Bhattacharyya coefficient, $BC(p, q) = \sum_{x \in X} \sqrt{p(x)q(x)}$. Choosing k and r thus becomes a search $\min_{k,r} BC(p, q)$.

Finally, after choosing k and r, we can simply choose θ as a value representing the 98th percentile (or similar) of the resulting distribution of random polygons, q. Practically, this value simply needs to separate p and q and visualization of these two distributions should suggest appropriate values.

We test this heuristic in Section 5.1 by comparing change detection performance from fitting our proposed model with this heuristic versus with labeled data.

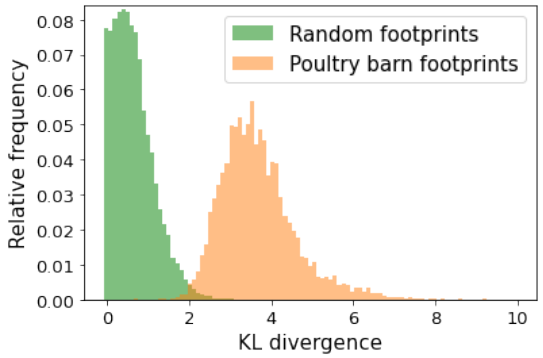

Figure 3: We show the distribution of KL divergence values generated by our proposed approach for (1) poultry barn footprints in aerial imagery where there is guaranteed to be a structure and (2) randomly generated footprints over similar aerial imagery. We observe that (1) is unimodal with a large mean value, as the color distributions of footprints that contain buildings are dissimilar to the color distribution of their surroundings and (2) is unimodal with a small mean value, as random patches are highly likely to have a color distribution that is similar to their neighborhood. Our proposed heuristic method looks to find hyperparameters for the model that minimize the overlap in these two distributions such that a simple threshold can identify changes in building footprints.

3.4 Baseline approaches

Here we propose a series of baselines and variants of our model to compare against. We refer to our proposed model / heuristic for fitting the model as "Semi-supervised TCM" as it only depends on labeled building footprints from a single point in time. This is specifically in contrast to variant approaches like "Supervised TCM"

(see below) that use labeled building footprints over time to fit the model parameters.

Supervised TCM Here, we fit the parameter θ in our proposed model using labeled data instead of our proposed heuristic. We can do this by searching over values of θ and measuring performance on the labeled data. In this case, k and r are model hyperparameters that can be searched over using validation data.

Supervised TCM with LR We use the series of KL-divergence values computed by TCM as a feature representation in a logistic regression (LR) model that directly predicts which point in time a structure is first observed. This is a supervised method as it requires a sample of labeled footprint data over time to fit.

Average-color with threshold This baseline uses the same structure as TCM with two changes: instead of clustering colors we compute average colors representations (over space for each spectral band) and instead of computing KL-divergence between distributions of cluster indices we compute the Euclidean distance between the average colors representations. Specifically, we compute the average color in a footprint and the average color of its neighborhood, then take the Euclidean distance between them and treat this distance in the same way we have previously treated the KL-divergence values. This has the effect of removing k as a hyperparameter, however the rest of the algorithm stays the same. Similar to the KL with threshold method we fit θ using labeled data.

Average-color with LR This method is identical to Supervised TCM with LR, but using the technique from Average-color with threshold to compute Euclidean distances between average color representations.

Color-over-time In this baseline we compute features from a time series of imagery by averaging the colors (over space for each spectral band) in the given footprint at each point in time, then taking the Euclidean distance between these average representations in subsequent pairs of imagery. For example, a time series of 5 images would result in an overall feature representation of 4 distances: the distance between the average colors at time 1 and average colors at time 2, the distance between the average colors at time 2 and the average colors at time 3, etc. We use this overall representation in a logistic regression model that predicts which point in time the structure is first observed.

CNN-over-time In this baseline we use the given structure footprints and satellite imagery at time t to train a U-Net based semantic segmentation model to predict whether or not each pixel in an image contains a developed structure. We then use this trained model to score the imagery from each point and time and determine the first layer in which a building is constructed. For simplicity, if the network predicts that over 50% of a footprint is constructed, then we count it as constructed.

Mode predictions This baseline is simply predicting the most frequent time point that we first observe constructed buildings based on the labels in the dataset of interest. For example, if '2011' was the most frequent year that we observed buildings to be constructed in a dataset, then this approach would predict every building was constructed in 2011 regardless of input. This serves as a lower bound on the performance of the supervised methods.

4 DATASETS

4.1 Poultry barn dataset

We use the Soroka and Duren dataset of 6,013 labeled poultry barn polygons, Poultry barn, created from aerial imagery from the National Agriculture Imagery Program (NAIP) from 2016/2017 over the Delmarva Peninsula (containing portions of Virginia, Maryland, and Delaware) [26]. NAIP imagery is 4 channel (red, green, blue, NIR) high-resolution ($\leq$ 1m/px) aerial imagery and is collected independently by each state in the US at least once every three years on a rolling basis. Because of this, the availability and quality of the imagery varies between states. For instance, the NAIP imagery from 2011 in Delaware and Maryland are collected on different days of the year, at different times of day, etc. See Figure 1 for example images of the NAIP imagery over time overlayed with the barn footprints. Additionally, we have manually labeled the earliest year (out of the years shown in Table 1) that a poultry barn can be seen for a random subset of 1,000 of the poultry barn footprints.

State	Years of NAIP data
Delaware	2011, 2013, 2015, 2017, 2018
Maryland	2011, 2013, 2015, 2017, 2018
Virginia	2011, 2012, 2014, 2016, 2018

Table 1: NAIP data availability over states covering the Delmarva Peninsula.

4.2 Solar farm dataset

We also use a solar farm dataset, Solar farm, containing polygons delineating solar installations in the Indian state of Karnataka for the year 2020. The dataset includes 935 individual polygons covering a total area or 25.7 km^2. The polygons were created by manually filtering the results of a model run on an annual median composite of Sentinel 2 multispectral surface reflectance imagery. We collect additional median composites of Sentinel 2 imagery for 2016 through 2019[2] to use for change detection. See Figure 1 for examples of the imagery overlayed with the solar farm footprints. For each of the 935 footprints we have manually labeled the earliest year (between 2016 and 2020) that a solar farm can be seen in the imagery.

5 EXPERIMENTS AND RESULTS

We experiment with different configurations of our algorithm on the Poultry barn and Solar farm datasets. In all experiments we measure the accuracy (ACC) – the percentage of labeled footprints

[2]The data from 2016, 2017 and 2018 are composites of the Sentinel 2 top of atmosphere products (Level 1C), while the 2019 and 2020 data are additionally corrected for surface reflectance (Level 2A). All data was processed with Google Earth Engine using the `COPERNICUS/S2` and `COPERNICUS/S2_SR` collections respectively.

	Method	Semi-Supervised	ACC	MAE
Poultry barn	Semi-supervised TCM	✓	0.94	0.15
	Supervised TCM		0.93 +/- 0.01	0.17 +/- 0.04
	Supervised TCM with LR		0.96 +/- 0.01	0.12 +/- 0.03
	CNN over time	✓	0.37	1.36
	Average-color with LR		0.95 +/- 0.01	0.15 +/- 0.05
	Average-color with threshold		0.91 +/- 0.02	0.24 +/- 0.06
	Color-over-time		0.90 +/- 0.02	0.41 +/- 0.08
	Mode predictions		0.84	0.80
Solar farm	Semi-supervised TCM	✓	0.71	0.49
	Supervised TCM		0.70 +/- 0.04	0.51 +/- 0.08
	Supervised TCM with LR		0.78 +/- 0.03	0.29 +/- 0.05
	CNN over time	✓	0.64	0.68
	Average-color with LR		0.65 +/- 0.03	0.49 +/- 0.05
	Average-color with threshold		0.50 +/- 0.04	0.93 +/- 0.08
	Color-over-time		0.79 +/- 0.01	0.29 +/- 0.02
	Mode predictions		0.42	0.81

Table 2: Comparison of our proposed semi-supervised model ("Semi-supervised TCM") to other baseline methods for detecting change in structures over time series of imagery. Note that the semi-supervised methods only have access to building footprint labels at time t, while the other methods are "supervised" and additionally have access to labels on when buildings were constructed over time. We observe that our semi-supervised approach achieves identical performance to a supervised variant where the model parameters are learned. We further observe the proposed approach outperforms supervised baseline methods for detecting change. Reported values are shown as averages (+/-) a standard deviation over 50 random train/test splits where appropriate. Single values are reported for the semi-supervised methods as they are evaluated on the entire labeled data set.

for which we correctly identify the first "developed" year and mean absolute error (MAE) – the average of absolute differences between the predicted year and labeled year.

5.1 Semi-supervised TCM

We first test how parameters chosen with the proposed heuristic correlate with performance of the model. The benefit of the heuristic method is that it does not require labeled temporal data to fit the model, but we need to show that the parameters it selects actually result in good performance. Here, we search over buffer sizes in $\{100, 200, 400\}$ meters and $\{0.016, 0.024\}$ degrees for the Poultry barn and Solar farm datasets, respectively, and number of clusters in $\{16, 32, 64\}$ for both datasets. For each configuration combination we create p and q as described in Section 3.3, compute the Bhattacharyya coefficient, estimate θ, then evaluate the predicted change years on the labeled data. We find that the Bhattacharyya coefficient is correlated with the result; there is -0.77 rank order correlation between the coefficient and accuracy (p=0.01) in Poultry barn and a -0.94 rank order correlation (p=0.004) in Solar farm. In both datasets, the smallest Bhattacharyya coefficient was paired with the best performing algorithm configuration.

Second, we compare the performance of the model with heuristic estimated parameters to that with learned parameters. To learn the parameters for our proposed model, "Supervised TCM", we randomly partition the labeled time series data into 80/20 train/test splits. We find the values of k, r, and θ (with a grid search over the same space for k and r as mentioned above) using the training split, then evaluate this model on the test split. We repeat this process for 50 random partitions and report the average and standard deviation metrics for the best combination Table 2. We observe that the heuristic method produces results that are equivalent to those of the learned model. In Poultry barn our proposed method achieves a 94% accuracy with a mean absolute error of 0.15 years which suggests it will be effective for driving longitudinal studies of the growth of poultry CAFOs.

Finally, we observe that our method significantly outperforms the other semi-supervised baseline, CNN over time. In both Poultry barn and Solar farm we observe considerable covariate shift. For example, in the Poultry barn dataset there is a large shift in input distribution over time due to the fact that the aerial imagery is collected at different days of the year, at different times of day, etc. The deep learning model is trained solely on imagery from the last point in each time series where we can confirm that there exists buildings in each footprint, however is unable to reliably generalize over time. We did not experiment with domain adaptation techniques to attempt to fix this, however we explore the use of our proposed method in this capacity in Section 6. We note that our proposed model is unaffected by shifts in the input distributions year-over-year as it never compares imagery from different years.

5.2 Supervised models

In the previous section we showed that we can estimate the parameters of our model without labeled time series data. Not requiring additional labeling is a major benefit of the TCM approach. In this section we explore the performance of our proposed approach

against *supervised* baseline approaches, using labels generated going back in time. We find that logistic regression models are effective at predicting the building construction date from the series of KL divergence values produced by our proposed approach (KL with LR). In both datasets this method is overall the top performing method with 96% accuracy in the Poultry barn dataset and 78% accuracy in the Solar farm dataset. In the Solar farm dataset the Color-over-time baseline has tied for top performance (within a standard deviation), but the same features are not as effective in the Poultry barn dataset where the color shifts are more dramatic year-over-year. Even so, the performance of the color-over-time baseline was much better than we originally hypothesized and should be compared to in future work regardless of perceived covariate shifts.

We also observe that our proposed method (that computes clustered representations) dominates the family of average-color baselines. This, along with the fact that we observe that more clusters in the k-means model usually results in better performance, suggests that the clustered representation is an important component of our approach. We hypothesize that more rich feature representations will prove even more effective as both the colors and textures of a footprint will change when it becomes developed. This is a trivial addition to the existing model and we hope to test it in future studies.

Finally, we observe that our heuristic method performs very well overall. In both datasets there are only two supervised methods that achieve stronger results than the semi-supervised proposed approach.

6 TEMPORAL CLUSTER MATCHING AS A DATA AUGMENTATION STRATEGY

In Table 2 we show that training semantic segmentation models on structure footprint masks at a time t does *not* result in a model that can generalize well over time (and thus cannot detect change). Previously (in Section 5.1) we hypothesized that this is due to the covariate shift in the time series imagery in the two datasets that we test on – the Poultry barn dataset uses NAIP aerial imagery that is collected at different times of day and different days of the year on a rolling three year basis and the Solar farm dataset uses Sentinel 2 mosaics created from TOA corrected imagery in 2016 through 2018 and surface reflectance corrected imagery in 2019 and 2020. Thus, a model trained with data from a single period in both of these cases is unlikely to perform well in other layers.

Here we test this hypothesis by using our proposed method to augment the data used to train the CNN over time method for the Poultry barn dataset. Specifically, we run Semi-supervised TCM over the Poultry barn dataset to create predictions as to when each footprint was constructed. We then use these estimates to create an expanded training set that contains pairs of imagery over all time points with footprint masks that are predicted to have a building. For example, if our model believes that there was a building in a given footprint at 2011 in the NAIP imagery, then we can train the segmentation model with (NAIP 2011, footprint), (NAIP 2013, footprint), etc. We find that this increases the performance of the model on the change detection task in all cases that we tested. For example, we apply this augmentation step to the same model configuration used in the results from Table 2 and achieve a 56% accuracy and 0.97 MAE (a 19% improvement in ACC and 0.39 improvement in MAE). These results are not competitive with the other methods we test in the change detection task. This likely stems from the fact that the segmentation model, in contrast to the other methods, is not specialized to change detection. On the other hand, the segmentation model can be run over new imagery to find novel instances of poultry barns (e.g., barns destroyed prior to the date of original data labeling), and is thus necessary to improve the performance of such models for more general applications.

While a more rigorous evaluation of how to improve the deep learning segmentation baseline is outside the scope of this paper, we hypothesize that more data augmentation strategies (e.g. RandAugment [4] and AugMix [13]), unsupervised domain adaptation methods [28], and a hyperparmeter search over dimensions such as class balancing strategies, temporal balancing strategies, learning rates, architecture, etc. would all improve performance. These types of experimentation will be critical for any future work that attempts to create general purpose models for detecting building construction at scale. That said, one of the main benefits of our proposed TCM is that it provides a lightweight approach to detect construction.

7 CONCLUSION AND FUTURE WORK

We have proposed Temporal Cluster Matching (TCM) for detecting change in building footprints from time series of remotely sensed imagery. This model is based on the intuition that the relationship between the distribution of colors inside and outside of a building footprint will change upon construction. We further propose a heuristic based method for fitting the parameters of our model and show that this approach effectively detects poultry barn and solar farm construction. TCM does not depend on having labels over time, yet it can outperform similar models that have such labels available. Further, we show that the feature representation from TCM – a sequence of KL-divergence values between the distribution of color clusters inside and outside of a building footprint – can be used in supervised models to improve change detection performance. Finally, we show how TCM can be used as a data augmentation technique for training deep learning models to *detect* building footprints from remotely sensed imagery.

This work motivates several future directions. First, the per-pixel representation of TCM will affect detectable changes. We used simple color representations, but more elaborate representations could be promising (e.g. texture descriptors or higher dimensional image embeddings). Second, future work should explore other applications of TCM. Here we experimented with imagery where the size of the footprints were relatively large compared to the spatial resolution of the imagery. However, our model may not perform as well when the footprint is relatively smaller. For example, we briefly experimented with detecting changes using general building footprints in the US and NAIP imagery and found that the relationship between the color distributions of small residential buildings and their surroundings was very noisy, although we did not attempt to investigate further. The top performing methods from the recent SpaceNet7 challenge run their building detection algorithms on upsampled imagery and a similar strategy may be useful with TCM.

Finally, we hypothesize that TCM would work with time series of imagery from multiple remote sensing sensor modalities. A benefit of our model is that it does not consider inter-year differences and thus is not affected by shifts in the color distributions of the imagery, but our experimental results do not explore the extent in which this is a useful property. Practically, there may be problems (such as shifts in geolocation accuracy) when applying TCM over stacks of imagery from different sources.

In summary, we hope TCM approach illustrated here will enable researchers to overcome the "frozen" labels of many emerging earth imagery datasets. Our lightweight approach to augment labels temporally should foster richer exploration of time series of satellite imagery and help us to understand the earth as it was, is, and will be.

ACKNOWLEDGMENTS

We thank Microsoft Azure for support in cloud computing, and Schmidt Futures, Stanford Impact Labs, and the GRACE Communications Foundation for research support.

REFERENCES

[1] Samaneh Aminikhanghahi and Diane J Cook. 2017. A survey of methods for time series change point detection. *Knowledge and information systems* 51, 2 (2017), 339–367.
[2] Jonathan Anomaly. 2015. What's Wrong with Factory Farming? *Public Health Ethics* (2015). https://ssrn.com/abstract=2392453
[3] Ying Chen, Xu Ouyang, and Gady Agam. 2019. ChangeNet: Learning to detect changes in satellite images. In *Proceedings of the 3rd ACM SIGSPATIAL International Workshop on AI for Geographic Knowledge Discovery*. 24–31.
[4] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. 2020. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*. 702–703.
[5] Ilke Demir, Krzysztof Koperski, David Lindenbaum, Guan Pang, Jing Huang, Saikat Basu, Forest Hughes, Devis Tuia, and Ramesh Raskar. 2018. Deepglobe 2018: A challenge to parse the earth through satellite images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 172–181.
[6] Adam Van Etten, Dave Lindenbaum, and Todd M. Bacastow. 2018. SpaceNet: A Remote Sensing Dataset and Challenge Series. *CoRR* abs/1807.01232 (2018). arXiv:1807.01232 http://arxiv.org/abs/1807.01232
[7] Michael J Evans and Jacob W Malcom. 2020. Automated Change Detection Methods for Satellite Data that can Improve Conservation Implementation. *bioRxiv* (2020), 611459.
[8] Anmar Frangoul. 2020. India has some huge renewable energy goals. But can they be achieved? *CNBC* (2020). https://www.cnbc.com/2020/03/03/india-has-some-huge-renewable-energy-goals-but-can-they-be-achieved.html
[9] Ananya Gupta, Elisabeth Welburn, Simon Watson, and Hujun Yin. 2019. CNN-Based Semantic Change Detection in Satellite Imagery. In *International Conference on Artificial Neural Networks*. Springer, 669–684.
[10] Rohit Gupta and Mubarak Shah. 2020. Rescuenet: Joint building segmentation and damage assessment from satellite imagery. *arXiv preprint arXiv:2004.07312* (2020).
[11] Cassandra Handan-Nader and Daniel E Ho. 2019. Deep learning to map concentrated animal feeding operations. *Nature Sustainability* 2, 4 (2019), 298–306.
[12] Cassandra Handan-Nader, Daniel E Ho, and Larry Y Liu. 2020. Deep Learning with Satellite Imagery to Enhance Environmental Enforcement. *Data-Driven Insights and Decisions: A Sustainability Perspective. Elsevier* (2020).
[13] Dan Hendrycks, Norman Mu, Ekin D Cubuk, Barret Zoph, Justin Gilmer, and Balaji Lakshminarayanan. 2019. Augmix: A simple data processing method to improve robustness and uncertainty. *arXiv preprint arXiv:1912.02781* (2019).
[14] Xin Huang, Yinxia Cao, and Jiayi Li. 2020. An automatic change detection method for monitoring newly constructed building areas using time-series multi-view high-resolution optical satellite images. *Remote Sensing of Environment* 244 (2020), 111802.
[15] Franck Jung. 2004. Detecting building changes from multitemporal aerial stereopairs. *ISPRS Journal of Photogrammetry and Remote Sensing* 58, 3-4 (2004), 187–201.
[16] Clemens Listner and Irmgard Niemeyer. 2011. Recent advances in object-based change detection. In *2011 IEEE International Geoscience and Remote Sensing Symposium*. IEEE, 110–113.
[17] José A Malpica, María C Alonso, Francisco Papí, Antonio Arozarena, and Alex Martínez De Agirre. 2013. Change detection of buildings from satellite imagery and lidar data. *International Journal of Remote Sensing* 34, 5 (2013), 1652–1675.
[18] Diego Marcos, Raffay Hamid, and Devis Tuia. 2016. Geospatial correspondences for multimodal registration. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 5091–5100.
[19] Masashi Matsuoka and Fumio Yamazaki. 2004. Use of satellite SAR intensity imagery for detecting building areas damaged due to earthquakes. *Earthquake Spectra* 20, 3 (2004), 975–994.
[20] Alex Moltzau. 2020. Estonia's National Strategy for Artificial Intelligence. *Medium* (2020). https://medium.com/swlh/estonias-national-strategy-for-artificial-intelligence-2623259ddf4c
[21] Sheila Moynihan. 2016. Mapping Solar Potential in India. *Office of Energy Efficiency and Renewable Energy* (2016). https://www.energy.gov/eere/articles/mapping-solar-potential-india
[22] David Osterberg and David Wallinga. 2004. Addressing Externalities From Swine Production to Reduce Public Health and Environmental Impacts. *American Journal of Public Health* 94, 10 (2004), 1703–1708. https://doi.org/10.2105/AJPH.94.10.1703 arXiv:https://doi.org/10.2105/AJPH.94.10.1703 PMID: 15451736.
[23] Ray Purdy. 2010. Using Earth observation technologies for better regulatory compliance and enforcement of environmental laws. *Journal of Environmental Law* 22, 1 (2010), 59–87.
[24] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*. Springer, 234–241.
[25] Ribana Roscher, Michele Volpi, Clément Mallet, Lukas Drees, and Jan Dirk Wegner. 2020. SemCity Toulouse: A benchmark for building instance segmentation in satellite images. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences* 5 (2020), 109–116.
[26] A.M. Soroka and Z. Duren. 2020. Poultry feeding operations on the Delaware, Maryland, and Virginia Peninsula from 2016 to 2017: U.S. Geological Survey data release. https://doi.org/10.5066/P9MO25Z7.
[27] Crista L. Straub, Stephen R. Koontz, and John B. Loomis. 2019. *Economic valuation of landsat imagery*. Technical Report. U.S. Geological Survey, Reston, VA. https://doi.org/10.3133/ofr20191112 Report.
[28] Yu Sun, Eric Tzeng, Trevor Darrell, and Alexei A Efros. 2019. Unsupervised domain adaptation through self-supervision. *arXiv preprint arXiv:1909.11825* (2019).
[29] A Tewkesbury. 2011. Mapping the extent of urban creep in Exeter using OBIA. In *Proceedings of RSPSoc Annual Conference*. 163.
[30] Andrew P Tewkesbury, Alexis J Comber, Nicholas J Tate, Alistair Lamb, and Peter F Fisher. 2015. A critical synthesis of remotely sensed optical image change detection techniques. *Remote Sensing of Environment* 160 (2015), 1–14.
[31] Devis Tuia, Claudio Persello, and Lorenzo Bruzzone. 2016. Domain adaptation for the classification of remote sensing data: An overview of recent advances. *IEEE geoscience and remote sensing magazine* 4, 2 (2016), 41–57.
[32] Adam Van Etten and Daniel Hogan. 2021. The SpaceNet Multi-Temporal Urban Development Challenge. *arXiv preprint arXiv:2102.11958* (2021).
[33] Adam Van Etten, Daniel Hogan, Jesus Martinez-Manso, Jacob Shermeyer, Nicholas Weir, and Ryan Lewis. 2021. The Multi-Temporal Urban Development SpaceNet Dataset. *arXiv preprint arXiv:2102.04420* (2021).
[34] Jan Verbesselt, Rob Hyndman, Glenn Newnham, and Darius Culvenor. 2010. Detecting trend and seasonal changes in satellite image time series. *Remote sensing of Environment* 114, 1 (2010), 106–115.
[35] Lei Wang, Congcong Li, Qing Ying, Xiao Cheng, Xiaoyi Wang, Xueyan Li, Luanyun Hu, Lu Liang, Le Yu, HuaBing Huang, et al. 2012. China's urban expansion from 1990 to 2010 determined with satellite remote sensing. *Chinese Science Bulletin* 57, 22 (2012), 2802–2812.
[36] Nicholas Weir, David Lindenbaum, Alexei Bastidas, Adam Van Etten, Sean McPherson, Jacob Shermeyer, Varun Kumar, and Hanlin Tang. 2019. Spacenet MVOI: a multi-view overhead imagery dataset. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 992–1001.
[37] Siyu Yang. 2018. How to extract building footprints from satellite images using deep learning. *Microsoft* (2018). https://azure.microsoft.com/en-us/blog/how-to-extract-building-footprints-from-satellite-images-using-deep-learning/
[38] Amy Zhang, Xianming Liu, Andreas Gros, and Tobias Tiecke. 2017. Building detection from satellite images on a global scale. *arXiv preprint arXiv:1707.08952* (2017).
[39] Kang Zhao, Jungwon Kang, Jaewook Jung, and Gunho Sohn. 2018. Building extraction from satellite images using mask R-CNN with building boundary regularization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 247–251.