Learning to Interpret Satellite Images in Global Scale Using Wikipedia

Burak Uzkent[1*], Evan Sheehan[1], Chenlin Meng[1], Zhongyi Tang[2], David Lobell[2], Marshall Burke[2], Stefano Ermon[1]

[1]Department of Computer Science, Stanford University
[2]Department of Earth Systems Science, Stanford University
buzkent@cs.stanford.edu, {esheehan, chenlin, zztang, dlobell, mburke}@stanford.edu,
ermon@cs.stanford.edu

Abstract

Despite recent progress in computer vision, fine-grained interpretation of satellite images remains challenging because of a lack of labeled training data. To overcome this limitation, we construct a novel dataset called WikiSatNet by pairing geo-referenced Wikipedia articles with satellite imagery of their corresponding locations. We then propose two strategies to learn representations of satellite images by predicting properties of the corresponding articles from the images. Leveraging this new multi-modal dataset, we can drastically reduce the quantity of human-annotated labels and time required for downstream tasks. On the recently released fMoW dataset, our pre-training strategies can boost the performance of a model pre-trained on ImageNet by up to 4.5% in F1 score.

1 Introduction

Deep learning has been the driving force behind many recent improvements in computer vision tasks, including image classification, image segmentation, object detection and tracking, etc. [21; 12; 4; 29; 27; 28]. These deep models, however, require training on high quality, large-scale datasets, and building these datasets is typically very costly. Satellite images are particularly difficult and expensive to label because of humans' unfamiliarity with aerial perspectives [1].

One effective way to reduce the amount of training data needed is to perform pre-training on an existing, previously annotated dataset, such as ImageNet [3], and transfer the learned weights to the domain of interest [17; 2; 30]. However, the success of this approach diminishes if the underlying distributions and/or compositions of the pre-training and target datasets are not sufficiently similar. Such a problem is exceptionally pronounced in the satellite imagery space, as the entire frame of reference and perspective of an aerial image is altered compared to a natural image. This has the unfortunate effect of rendering natural image datasets, such as ImageNet, less useful as pre-training mechanisms for downstream computer vision tasks in the satellite domain [16; 7].

*Contact Author

Because direct annotation is expensive, researchers have considered many creative ways to provide supervision without explicit labels. These include unsupervised [9], label-free [19; 25], and weakly supervised learning methods [18]. A particularly effective strategy is to leverage co-occurrence statistics in a dataset, e.g., predict the next frame in a video, a missing word in a sentence [14], or predict relationships between entities such as images and text co-occurring together. For example, leveraging images and their hashtags on Instagram, [13] build a large scale image recognition dataset consisting of more than 3 billion images across 17,000 weak labels obtained from textual hashtags and their WordNet [15] synsets. After pre-training on this extremely large dataset, they report almost 5% improvement over the same model trained from scratch on ImageNet.

Because satellite images are geolocated, i.e., they correspond to specific locations (and times), they can be paired with other geolocated datasets (e.g., OpenStreetMap [7]), exploiting spatial co-occurrence statistics as a source of supervision [23; 22]. Following this strategy, we construct a novel multi-modal dataset by pairing geo-referenced Wikipedia articles with their corresponding satellite images. By treating an article as an information-rich label, we obtain highly detailed physical and qualitative context for each image. For example, the first sentence of the John. F. Kennedy International Airport article contains excerpts such as "*JFK is the primary international airport serving New York City*". Wikipedia articles additionally contain demographic, environmental, and social information in structured form. To the best of our knowledge, this is the first time that Wikipedia has been used in conjunction with satellite images, and with 888,696 article-image entries, our approach yields the *largest satellite image dataset* to date.

In this paper, we demonstrate the effectiveness of pairing Wikipedia articles to satellite images for pre-training CNNs for satellite image recognition. We propose two pre-training methods to learn deep representations. First, similar to [13], we weakly label satellite images with curated summarization tags extracted from the article via an automated process. We then train a deep convolutional network to predict these weak labels directly from the images, learning useful representations in the process. In the second approach, we propose a novel joint architecture where we first obtain a textual embedding of each article using document summarization tech-

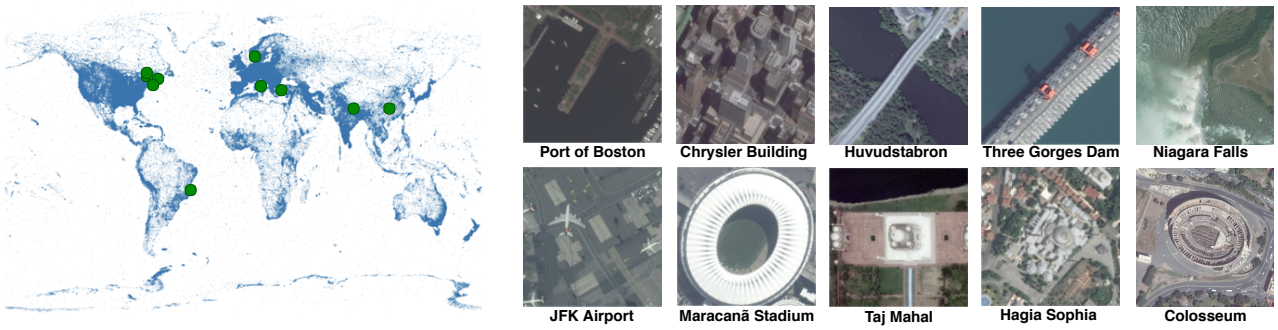


Figure 1: **Left**: Scatter plot of the distribution of geo-tagged Wikipedia articles together with some images (**right**) matched to the articles shown as green dots on the left plot. The title of the Wikipedia articles are written under each image. Zooming-in is recommended for visualization.

niques from NLP [10] and then train a deep convolutional network to produce an embedding for each image that is "similar" to the textual one. The first approach is a crude way of getting a single weak label for each article whereas the second learns representations without weak labels. The pre-trained networks are then evaluated on a downstream hand-labeled dataset, as in [6], where we obtain 4.5% higher accuracy compared to networks pre-trained on ImageNet, the standard approach for computer vision tasks.

2 Pairing Rich Crowdsourced Annotations from Wikipedia to Satellite Images

Wikipedia is a large-scale, crowdsourced database spanning 302 languages with over 47 million articles [32]. Of these 47 million articles, about 11% are contained in the English version. Out of these approximately 5 million articles, we found that roughly 1 million, or nearly 20%, are geolocated, meaning there is a latitude and longitude $c_i = \{c_i^{lat}, c_i^{lon}\}$ associated with the article's text y_i. Our *key idea is to use the article's coordinates to acquire a satellite image of its location from space* (see Fig. 1).

There is often a strong correlation between the article's text, y_i, and the visual content of the corresponding image, x_i. Indeed, we can think of the article as an extremely detailed "caption" for the satellite image, providing an often comprehensive textual representation of the satellite image, or an *information-rich label*. This label often contains structured data in the form of tables, called infoboxes, as well as raw text, allowing for the extraction of information about the physical state and features of the entity (e.g., elevation, age, climate, population).

2.1 Acquiring Matching Satellite Imagery

For a given article's coordinate c_i, there are many sensors that can provide imagery, with different tradeoffs in terms of spatial and temporal resolution, wavelengths, and costs. In this paper we acquire high resolution images from DigitalGlobe satellites. The images have a ground sampling distance (GSD) of 0.3-$0.5m$. These are among the highest resolution images available commercially, and were also used in the recently released functional map of the world (fMoW) dataset [1]. Note that one could also use the same strategy to build a similar multi-modal dataset using lower-resolution (10 meter), publicly available Landsat and Sentinel-2 images. For a given coordinate c_i, there are usually multiple images available, captured at different times. We acquired the latest image available. Another important design choice is the size of the acquired images. In this study, we use 1000×1000 pixels images covering approximately an area of $900m^2$. In aerial images, objects occupy drastically different numbers of pixels, as shown in Fig. 1. Based on preliminary manual examination, we found that 1000×1000 pixels images can typically cover most of the relevant objects. Finally, we prioritized collecting RGB images and only acquired grayscale images if an RGB image was not available. We did not perform any filtering to remove cloudy images, as our goal is to learn robust representations on a noisy dataset.

Our resulting *WikiSatNet* multi-modal dataset is a set of tuples $\mathcal{D} = \{(c_1, x_1, y_1), (c_2, x_2, y_2), \cdots, (c_N, x_N, y_N)\}$ where each tuple (c_i, x_i, y_i) represents a location (c_i), corresponding DigitalGlobe image (x_i) and Wikipedia article text (y_i). *WikiSatNet* contains $N = 888,696$ article-image pairs. To the best of our knowledge, this is *the largest dataset to date consisting of satellite images* and about 2 times larger than the recently released large scale fMoW dataset. Note that our procedure is highly scalable and fully automated. It could be used to generate even larger datasets by considering other Wikipedia languages and other sensors in addition to DigitalGlobe. In the next section, we propose two novel methods to pre-train a convolutional neural network (CNN) to extract information about images x_i using information from y_i.

3 Learning Visual Representations using Wikipedia Textual Information

Exemplifying the diverse application possibilities highlighted in the previous sections, we construct a general Wikipedia article-satellite image framework for pre-training CNNs. We then explore whether we can learn to interpret satellite images using knowledge extracted from Wikipedia articles via two approaches: weakly-supervised [18] labelling and a novel textual embedding method that attempts to match textual and visual embeddings.

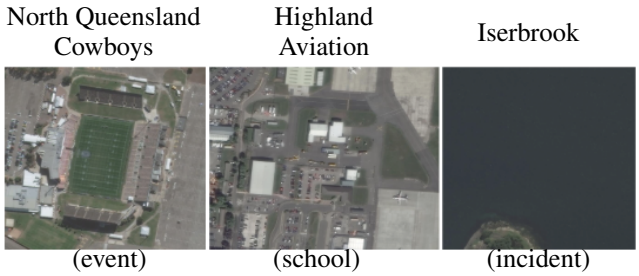


Figure 2: Some of the extracted weak labels representing *flipped* label noise. Corresponding Wikipedia article titles are written above the images. Though the words *stadium*, *airport*, and *water* are mentioned 19, 6, and 23 times in the articles, our weak label extraction pipeline generates wrong labels. Using image to text matching helps alleviate this flipped label noise.

Figure 3: Visually similar examples where the extracted weak labels cause *adversarial* label noise. Here the CNN is penalized for errors even when the predicted label is visually similar to assigned weak label. In contrast, our document summarization model projects the embeddings of the articles of these images to a similar space to avoid penalizing the CNN when predicting a similar label.

3.1 Weakly Supervised Learning

We first propose learning visual features using a data-programming pipeline [18] to label our dataset. We begin by extracting a weak label $\hat{w}(y_i)$ for each article y_i in our dataset. In our context, a weak label is a noisy, machine-generated classification of an article from a set of pre-defined labels. Because of space constraints, we only provide a high-level description of the approach, and will add more details by purchasing extra pages in the final version. As a first step, we manually compile a list of 97 potential categories that an article could fall under (e.g., $city$, $lake$, $event$, etc.) and use regular expressions to search for the terms throughout specific areas of the article's text where article meta-data is contained. We then rank the categories which are matched to the article in a manually-constructed hierarchical fashion from specific to general (e.g., $building \longrightarrow town \longrightarrow county$, etc.) and choose the one which comes first to label the article. Because many of these category labels are very detailed, we then merge certain similar categories together to create more general labels. We also discard articles that are assigned labels which cannot be determined from a satellite image (e.g., $person$, $event$, etc.). Weak labels represented by less than 100 samples are also removed, reducing the final set of labels to 55.

Given the final set of weak labels and corresponding images, we train a classifier to predict $\hat{w}(y_i)$ from x_i. The classifier is composed of a convolutional neural network $f_v : \mathcal{X} \mapsto \mathbb{R}^M$ that embeds images into an M dimensional feature space, followed by fully connected and softmax layers as shown in Fig. 4a. In this study, we parameterize f_v using the DenseNet121 [5] architecture which was previously shown to perform well across a range of tasks. The classifier is trained using the cross entropy loss function. The features learned by the convolutional embedding f_v on this large-scale pre-training task can then be transferred to downstream tasks, e.g., object detection or land cover classification.

Extracting weak labels is a noisy process that leads to a significant number of *flipped* labels as shown in Fig. 2. Additionally, the process leads to *adversarial* label noise because of visually similar labels such as *city, country, populated place, building, town* etc., as shown in Fig. 3. One can apply a simple merging step to place such visually similar labels into a general category, e.g., *populated place*. However, it leads to a class imbalance problem where almost 40% of the dataset is dominated by populated places. Exploring the trade-off between adversarial label noise and class imbalance problems is a very time-consuming process due to the nature of working with a large-scale dataset. For this reason, in the next section, we propose a novel, and practical method to learn deep representations using multi-modal data without manual preprocessing.

3.2 Image to Text Matching Learning

In this section, we propose a novel method to learn deep convolutional features without using hand-crafted labeling functions. This not only substantially reduces human effort, but also tackles the adversarial label noise by softening the loss function for the images that can fall into multiple visually similar categories. Our method relies on the idea of image to text matching [11; 31]. In this direction, we propose a novel network shown in Fig. 4b with two branches: a *visual* and a *textual* one. We design a loss function that encourages the CNN (*visual* branch) to produce image embeddings that are close to a suitable vector representation of the corresponding article's text (*textual* branch).

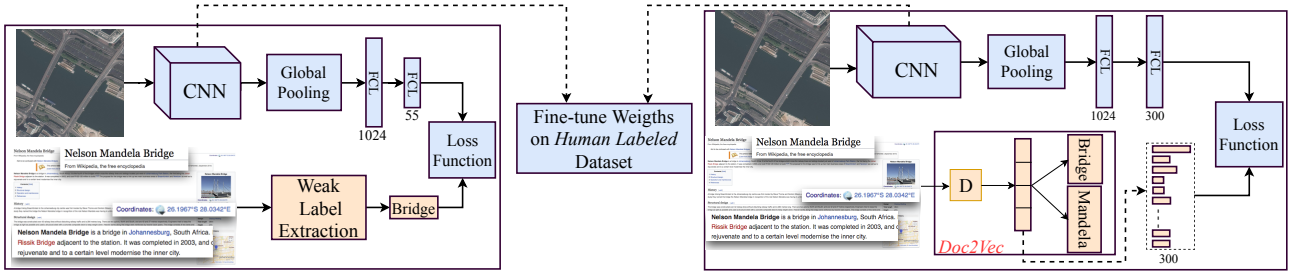
The proposed architecture uses satellite images, $\mathcal{X}$, and Wikipedia articles, $\mathcal{Y}$, as input. In the *textual* branch, we learn a function $f_t : \mathcal{Y} \mapsto \mathbb{R}^K$, to project an article, y_i, to a textual embedding space $z_i^t \in \mathbb{R}^K$ using a document summarization model from natural language processing (NLP):

$$z_i^t = f_t(y_i). \tag{1}$$

In the *visual* branch, we use a function $f_v : \mathcal{X} \mapsto \mathbb{R}^M$ parameterized using a convolutional neural network to extract features from an image as

$$z_i^v = f_v(x_i) \tag{2}$$

where i represents the index of the image paired to article y_i. We parameterize f_v using the DenseNet121 architecture [5] as in the weak supervision method. Next, we use a function $f_m : \mathcal{Z}^v \mapsto \mathbb{R}^K$ to map z_i^v to the same dimension as the textual feature vector z_i^t. The function f_m is parameterized using a fully connected layer with ReLU activations. The final feature vectors, z_i^v and $z_i^t \in \mathbb{R}^K$, are then compared with a loss function that enforces similarity.

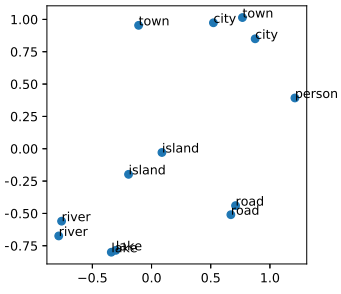


(a) (b)

Figure 4: The workflow of the proposed weakly supervised learning method (a): (1) Extract labels from articles using our labeling pipeline. (2) Match articles with images of their coordinates. (3) Pre-train on a large-scale dataset using 55 weak labels. (4) Transfer learned weights to a down-stream task. In (b) we show the workflow of the image to text matching learning. Our method enforces the CNN to learn features similar to raw textual features learned by *Doc2Vec*.

Pre-training the *Doc2Vec* Model

Our image to text matching method uses textual descriptors $\mathcal{Z}^t$ to learn deep visual representations. In our study, we use the *Doc2Vec* network [10] which can summarize variable length articles in a unified framework. *Doc2Vec* is a document summarization method that can take a variable length piece of text, y_i, and map $y_i \in \mathcal{Y}$ to a paragraph vector $z_i^t = f_t(y_i) \in \mathbb{R}^K$ in a fixed-length vector space, where K is specified by the user. Documents that possess similar meanings are mapped to nearby points in the embedding space, allowing a comparison between any two documents. In contrast to fixed length vector representations using Bag-of-Words, *Doc2Vec* can capture the orderings and semantics of the words, which is highly beneficial for our unsupervised learning task. For example, learning a textual embedding space where we can closely map article categories such as *country*, *city*, *town* etc. is desired considering that their corresponding visual data contain similar structures (see Fig. 5). Another advantage of the *Doc2Vec* model is that it is an unsupervised learning model. This allows us to learn Wikipedia-specific descriptors by training it on the full geolocated Wikipedia article corpus.



City - Middletown, Connecticut
City - Milton, Georgia
Lake - Timothy Lake
Lake - Tinquilco Lake
Town - Mingona Township, Barber County, Kansas
Town - Moon Township, Allegheny County, Pennsylvania
Road - Morehampton Road, Dublin
Road - Motorway M10 Pakistan
River - Motru River
River - Mousam River
Island - Aupaluktok Island
Island - Avatanak Island

Figure 5: Visualization of PCA components of the randomly chosen articles learned by *Doc2Vec*. Notice that visually similar objects such as *city*, *town* are closely mapped while different objects are projected far away. Corresponding Wikipedia article titles are shown on the right.

Cosine Similarity Loss Function

After learning feature vectors, z_i^v and $z_i^t \in \mathbb{R}^K$, from the two branch network, we apply a loss function to measure the similarity of the two vectors. We propose using the cosine similarity metric, which measures the angle, θ_i, between two vectors as

$$D(x_i, y_i) = cos(\theta_i) = \frac{f_v(x_i)^T f_t(y_i)}{\|f_v(x_i)\|_2 \, \|f_t(y_i)\|_2}. \quad (3)$$

Wikipedia has varying lengths of articles, which makes the cosine similarity function ideal since it measures the similarity between the direction rather than the magnitude of two vectors.

One can apply some other loss functions for our pre-training task. For example, [31] proposed triplet loss function where the anchor is a phrase paired to its corresponding positive visual data. The negative image is then sampled from the neighborhood of the positive sample. [6] adapted triplet loss function for unsupervised learning on satellite images. They assign a positive image for each anchor image from its spatial neighborhood following the assumption that nearby images contain similar visual structures. The negative sample is then sampled from the areas outside the anchor's neighborhood circle. In our case, we lack explicit knowledge that can help us sample negative image given an article, y_i, as anchor and its corresponding image, x_i, as positive. In this direction, one can compute the similarity in the visual, z_{v1}, or textual, z_t, embedding space between a positive sample and other samples in a certain spatial neighborhood to get a negative sample.

Another interesting aspect of our architecture is the dimensionality of the textual embeddding space. We believe that 300 dimensional feature vector can capture all the fine-grained visual structures in an aerial image. However, during our experiments we observed that visually similar features can lead to more uniform textual descriptors slowing down the learning process. Thus, using a smaller dimensional embedding space can lead to more discriminative visual features that can potentially speed up the learning process. On the other hand, this can also prevent the CNN from learning fine-grained in- formation. We leave the task of exploring this trade-off as a future work of our study. Another future

dimension of our image to text matching work is generating a Wikipedia article, y_i, using an NLP decoder $f_t^{'}$ given visual representations x_i as

$$y_i = f_t^{'}(f_v(x_i)). \quad (4)$$

Training on *WikiSatNet*

In our pre-training experiments, we use similar hyper-parameters in both weak supervision and image to text matching to train the DenseNet121 for optimizing the weights for f_v. We initialize weights randomly, however, we observed faster convergence when initializing with pre-trained weights. After experimentation, we set the learning rate and batch size to 0.0001 and 64, respectively, and the Adam optimizer is used to train the model [8]. Finally, we resize the 1000×1000 pixels images to 224×224 pixels images to compare with publicly available datasets.

Figure 6: Some of the cloudy images in WikiSatNet. The cloudy images amount to roughly 5-7% of the dataset.

In the initial steps of image to text training, we observe an angle of approximately 90^o ($D(x_i, y_i) \approx 0$) between z_i^t and z_i^v. This is consistent with the fact that random vectors in high dimensional spaces are likely to be orthogonal to each other. After several epochs, the angle decreases to about 45^o ($D(x_i, y_i) \approx 0.5$) and stops decreasing further. We believe that this is partially due to articles that do not contain any visual cue, e.g *culture* and *person*, and also cloudy images (see Fig. 6), which amount to roughly 5% of the dataset. We did not observe over-fitting in our experiments. While we are not able to achieve zero loss, we qualitatively find that our approaches learn meaningful representations. To verify this, after pre-training the CNN on *WikiSatNet* using the image to text matching, we visualize the cosine similarities between z_i^t and z_i^v as shown in Fig. 7. In the same figure, we keep z_t fixed and use embeddings from images at different locations. The CNN learns to project embedding z_i^v closer to its corresponding article embedding z_i^t. We will publicly release the code for our image to text matching and weak supervision methods upon publication. Additionally, we expect to release a substantial fraction of the high resolution images in *WikiSatNet* (negotiations on the license with the image provider are ongoing). This will encourage further research into jointly utilizing Wikipedia and satellite images.

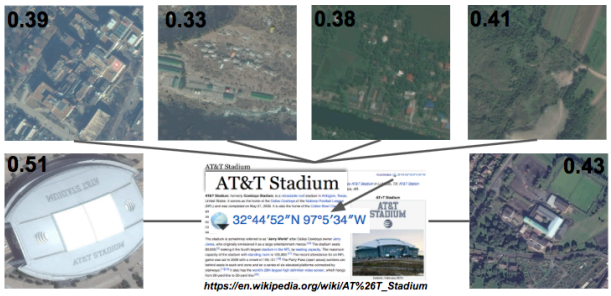


Figure 7: Visualization of cosine similarities learned by the CNN. The cosine similarities between the CNN embeddings and the *Doc2Vec* embedding are computed and overlaid on the images. The CNN learns to embed AT&T Stadium's image closer to the its corresponding article.

4 Transfer Learning Experiments

After pre-training CNNs on *WikiSatNet* using the proposed methods, we test them on three target tasks: (1) single image classification on the fMoW dataset, (2) temporal view classification using multiple images over an area on the fMoW dataset, and (3) land cover classification. In these tasks, we compare our pre-training strategies to the following baselines: (1) pre-training on ImageNet [21], (2) pre-training on CIFAR10, and (3) training from scratch. Our goal is to evaluate whether we learn satellite-specific representations that outperform the ones obtained using out-of-domain benchmarks with human labels.

Fine-tuning

There are two classical approaches in fine-tuning a deep network on the target task: (1) training all layers, and (2) freezing all the layers other than the final classification layer. In our experiments, we present results from both strategies. The learning rates for the weakly supervised and image to text matching model are set to 1e-4 and 1e-5 after experimentation. On the other hand, the learning rate for the ImageNet model is set to 1e-4, while it is set to 1e-3 for both the CIFAR10 pre-trained and trained from scratch models. These were the best performing hyper-parameters in our experiments. Finally, resized 224×224 pixel RGB images are used as input to the model as in the pre-training task. We follow the same approach for the models pre-trained on CIFAR10 and ImageNet.

4.1 Experimenting on the fMoW Dataset

To quantify the quality of the representations learned in the pre-training step, we first use a recently released large-scale satellite image recognition dataset named fMoW [1]. The fMoW dataset consists of both multispectral and RGB images and contains 83,412 unique training bounding boxes from large satellite images representing 62 different objects. The validation and test sets contain 14,241 and 16,948 bounding boxes and are left unchanged in our experiments. It also comes with temporal views from the same scenes, making classification of some classes such as *construction site* and *flooded road* easier. [1] proposes a multi-modal architecture that uses a DenseNet161 pre-trained on ImageNet and an

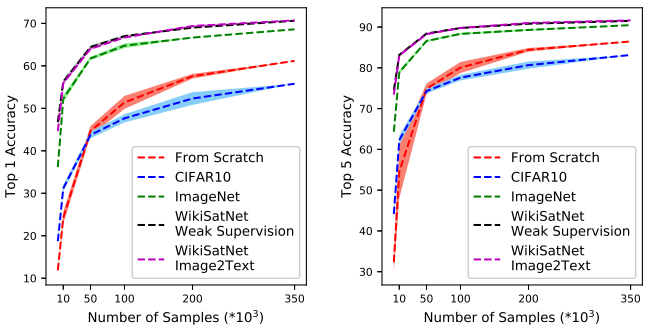


Figure 8: The top-1 and 5 classification accuracies of the proposed pre-training and baseline strategies on fMoW's test set when fine-tuning all layers on fMoW's training set. Monte-Carlo experiments were conducted when sampling a subset of the full training set.

LSTM to learn from images and their corresponding meta-data. Their DenseNet161 model has a number of parameters similar to the DenseNet121 model we use in our experiments. Since our pre-training framework learns from visual data, it can be easily applied to any CNN model to boost performance as well as reduce the number of labeled samples needed for a target task.

Reasoning on Single Images

In the first task, we perform experiments on the fMoW dataset for the task of classifying individual images using features extracted by the *visual* branch $f_v(\cdot)$ as

$$L(x_i) = \underset{j}{\text{argmax}}(p(f_v(x_i))) \tag{5}$$

where p represent the fully connected and softmax layers whereas j denotes the index of the assigned label. We experiment with 2000, 10000, 50000, 100000, 200000, and 350000 training images. As shown in Fig. 8, our pre-training strategies outperform the other pre-training strategies by large margins in top-1 and top-5 classification accuracy when using small amounts of labeled data. For example, when using 2000 labeled images, both our training strategies outperform ImageNet and CIFAR10 by 10% and 30%, respectively. As expected, this number goes down to about 5% and 20% when increasing the number of labeled images to 50000. Interestingly, at this point, the model trained from scratch starts to outperform the model pre-trained on CIFAR10. When using the full training set, our proposed pre-training strategies outperform ImageNet by about 2% and outperform the model trained from scratch by about 10%. These results demonstrate that our proposed approach produces features that are highly beneficial in down-stream tasks involving satellite images, even when large numbers of human labeled samples are available. When fine-tuning only the final layer, the proposed pre-training methods outperform ImageNet features by about 13% on the test set as shown in Table 1.

Model	CIFAR10	ImageNet	WikiSatNet *Weak Labels*	WikiSatNet *Image2Text*
Top-1 Acc. (Fixed f_v)	13.98 (%)	37.73 (%)	50.73 (%)	**51.02** (%)
Top-1 Acc. (Fine-tuned f_v)	55.79 (%)	68.61 (%)	70.62 (%)	**70.72** (%)

Table 1: Top-1 accuracies on the fMoW test set for pre-trained models. All the models are fine-tuned on the full fMoW training set. Fixed f_v represents the fine-tuning method where the pre-trained weights are fixed whereas the second method fine-tunes all the layers.

Reasoning on Temporal Views

In this section, we evaluate our representations on the task of temporal view classification across 62 classes from the fMoW dataset. This way, we can understand if our pre-training methods also boost performance on tasks that use temporal data as input. [1] trains the network on single labeled images and at test time averages the softmax predictions of the network on different images from the same area to assign the label with the maximum average score. We follow their training and test methods and at test time average predictions from T images over the same area, again using features extracted from $f_v(\cdot)$ as input. This can be formulated as

$$L(X) = \underset{j}{\text{argmax}}(mean(\sum_{t=1}^{T} p(f_v(x_t)))) \tag{6}$$

where j denotes the index of the assigned label and f_v represents the pre-trained network fine-tuned on the fMoW. Different from the previous section, we now report results in F1-scores to compare our models to the ones proposed by [1].

Model	CIFAR10	ImageNet	WikiSatNet *Weak Labels*	WikiSatNet *Image2Text*
F1 Score (*Single View*)	55.34 (%)	64.71 (%)	66.17 (%)	**67.12** (%)
F1 Score (*Temporal Views*)	60.45 (%)	68.73 (%)	71.31 (%)	**73.02** (%)

Table 2: F1 scores of different pre-training methods on fMoW's test set when fine-tuning all the layers on fMoW's training set.

We first compare our pre-training methods to ImageNet and CIFAR10 pre-training in Table 2. The proposed pre-training methods outperform the ImageNet pre-trained model by up to 4.5% in F1 Score when performing reasoning on temporal views. Among the proposed methods, the image to text matching approach outperforms the weak supervision with handcrafted labels method by about 1.7% in F1 Score. These results prove that the importance of pre-training does not diminish when switching from single to temporal views. On the other hand, [1] proposes five different models for the fMoW classification task. Three of them use meta-data and images jointly, whereas the remaining two only employ an ImageNet pre-trained DenseNet on images. Their visual data-only models are named *CNN-I-1* and *CNN-I*, where the former is a single view model and the latter performs tempo-

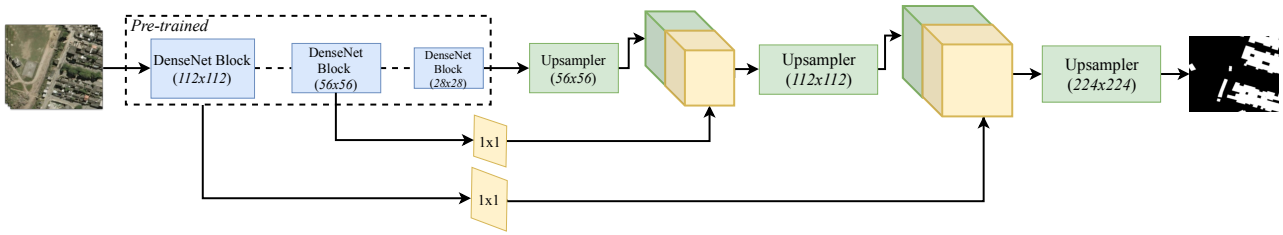


Figure 9: The proposed segmentation architecture that uses the pre-trained weights in the encoder stage.

ral reasoning. We can improve these models with our pre-training strategy by about 4.5% in F1 score while performing similarly to their top performing model, *LSTM-IM*, which uses meta-data and visual data jointly to perform temporal reasoning. Although this is outside the scope of this paper, our pre-trained models can replace the DenseNet model, pre-trained on ImageNet, used in *LSTM-IM* to improve its results as well.

Our experiments demonstrate that pre-training with weak or no supervision is very useful for the target task as reported by [13] both in terms of (1) boosting accuracy and (2) reducing the required human labeled dataset size on the target task. Unlike the pre-training framework proposed by [13], we do not necessarily need to have billions of images to overcome noise to learn useful representations.

4.2 Experiments on Land Cover Classification

Additionally, we perform classification across 66 land cover classes using remote sensing images with $0.6m$ GSD obtained by the USDA's National Agriculture Imagery Program (NAIP). We focus on the images from the California's Central Valley near the city of Fresno for the year 2016. The corresponding land cover map, named the Cropland Data Layer (CDL), is collected by the USDA for the continental United States [26]. The CDL is provided at $30m$ GSD, and we upsample them to match $0.6m$ GSD to use as ground truth. The final dataset consists of 100000 training and 50000 validation and test images. We only fine-tune the classification layer while keeping f_v fixed.

Model	CIFAR10	ImageNet	WikiSatNet *Weak Labels*	WikiSatNet *Image2Text*
Top 1 Acc.	42.01 (%)	40.11 (%)	46.16 (%)	**47.65** (%)
Top 5 Acc.	74.73 (%)	80.15 (%)	88.66 (%)	**88.77** (%)

Table 3: Performance of different pre-training methods on the land cover classification task.

As shown in Table 3, our pre-training strategies lead to substantially higher performance than the ImageNet and CIFAR10 features. This demonstrates the robustness and wide range of applications our pre-training strategies possess.

4.3 Experiments on Semantic Segmentation

Previously, we explored image recognition in both pre-training and target tasks. In this section, we change the target task type to semantic segmentation to understand if image recognition pre-training can still boost the performance on semantic segmentation target task. Image recognition focuses on global features to associate an image x_i with a label w_i. On the other hand, in semantic segmentation, local features play more important role in associating a pixel $x_i(m, n)$ with a label w_i where m and n represent the column and row index.

We first build a semantic segmentation network using the DenseNet121 model pre-trained on WikiSatNet. A typical segmentation model consists of encoder-decoder steps to generate segmentation maps with the same size to input images. In this case, we use the pre-trained weights f_v as encoder and design a decoder architecture on top of f_v as shown in Fig. 9. The proposed architecture is similar to U-Net architecture [20], however, we use the DenseNet121 pre-trained on WikiSatNet in the decoder stage. This is important as it requires a complex network to learn from large-scale datasets such as WikiSatNet. On the other hand, the U-Net employs a shallow encoder, preventing us to pre-train it on the WikiSatNet. We perform experiments on the SpaceNet [24] semantic segmentation task on the satellite images. The SpaceNet contains training and test set from six different cities, and building and road masks for corresponding high resolution (0.3-0.5m GSD) DigitalGlobe images. In this study, we focus on the *Rio* set and building masks. There are about 5000 training and 800 test images coming from the city of Rio de Janeiro. We experiment with varying number of training samples to quantify the learned representations in the case of using different amount of labeled samples in the target task. However, we keep the test set unchanged in our experiments. Table 4 shows the Intersection-over-Union (IoU) scores of the proposed segmentation architecture (see Fig. 9) when pre-trained on ImageNet and WikiSatNet.

Figure 10: An example of a building mask for a satellite image in SpaceNet *Rio* dataset.

As shown in Table 4, the pre-training provides significant boost when fine-tuning on small amount of training samples

Model	From Scratch	ImageNet	WikiSatNet *Image2Text*
200 Samples	42.11 (%)	50.75 (%)	**51.70** (%)
500 Samples	48.98 (%)	54.63 (%)	**55.41** (%)
5000 Samples	57.21 (%)	59.63 (%)	**59.74** (%)

Table 4: The IoU scores of different pre-training methods on building segmentation task.

(200, and 500 samples). However, pre-training on the WikiSatNet only achieves slightly higher IoU than ImageNet. These results are consistent with the previous studies where pre-training and target datasets contain different level tasks [13]. For example, [13] explored the idea of pre-training on image recognition task and transferring the learned weights for the task of object detection. They report that such set up does not lead to significant performance increase as in the case where both pre-training and target tasks are the same-level tasks (image recognition).

5 Conclusion

In this study, we proposed a novel combination of satellite images and crowdsourced annotations from geo-referenced Wikipedia articles. To the best of our knowledge, this is the first time that Wikipedia has been used this way. Our approach yields a large scale, multi-modal dataset combining rich visual and textual information for millions of locations all over the world — including additional languages beyond English will likely improve coverage even more. Leveraging paired multi-modal data, we proposed two different pre-training methods: (1) learning with weak labels, and (2) learning without weak labels using image to text matching. Both pre-training strategies lead to improved results on the recently released fMoW dataset consisting of large numbers of labeled samples. Our image to text matching model outperformed one pre-trained on ImageNet by 4.5% when using around 350000 labeled samples; this increase in performance is substantially higher when there are fewer labeled samples available.

References

[1] Gordon Christie, Neil Fendley, James Wilson, and Ryan Mukherjee. Functional map of the world. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, Utah*, 2018.

[2] Wenyuan Dai, Ou Jin, Gui-Rong Xue, Qiang Yang, and Yong Yu. Eigentransfer: a unified framework for transfer learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 193–200. ACM, 2009.

[3] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. Ieee, 2009.

[4] Junwei Han, Dingwen Zhang, Gong Cheng, Nian Liu, and Dong Xu. Advanced deep-learning techniques for salient and category-specific object detection: a survey. *IEEE Signal Processing Magazine*, 35(1):84–100, 2018.

[5] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2261–2269. IEEE, 2017.

[6] Neal Jean, Sherrie Wang, George Azzari, David Lobell, and Stefano Ermon. Tile2vec: Unsupervised representation learning for remote sensing data. *arXiv preprint arXiv:1805.02855*, 2018.

[7] Pascal Kaiser, Jan Dirk Wegner, Aurélien Lucchi, Martin Jaggi, Thomas Hofmann, and Konrad Schindler. Learning aerial image segmentation from online maps. *IEEE Transactions on Geoscience and Remote Sensing*, 55(11):6054–6068, 2017.

[8] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[9] Diederik P Kingma, Shakir Mohamed, Danilo Jimenez Rezende, and Max Welling. Semi-supervised learning with deep generative models. In *Advances in Neural Information Processing Systems*, pages 3581–3589, 2014.

[10] Quoc Le and Thomas Mikolov. Distributed representations of sentences and documents. *arXiv preprint arXiv:1405.4053*, 2014.

[11] Jimmy Lei Ba, Kevin Swersky, Sanja Fidler, et al. Predicting deep zero-shot convolutional neural networks using textual descriptions. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4247–4255, 2015.

[12] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[13] Dhruv Mahajan, Ross Girshick, Vignesh Ramanathan, Kaiming He, Manohar Paluri, Yixuan Li, Ashwin Bharambe, and Laurens van der Maaten. Exploring the limits of weakly supervised pretraining. *arXiv preprint arXiv:1805.00932*, 2018.

[14] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781, 2013.

[15] George A Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.

[16] Sinno Jialin Pan, Qiang Yang, et al. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2010.

[17] Rajat Raina, Alexis Battle, Honglak Lee, Benjamin Packer, and Andrew Y Ng. Self-taught learning: transfer learning from unlabeled data. In *Proceedings of the 24th international conference on Machine learning*, pages 759–766. ACM, 2007.

[18] Alexander Ratner, Stephen H. Bach, Henry Ehrenberg, Jason Fries, Sen Wu, and Christopher R. Snorkel: Rapid training data creation with weak supervision. *arXiv preprint arXiv:1711.10160*, 2017.

[19] Hongyu Ren, Russell Stewart, Jiaming Song, Volodymyr Kuleshov, and Stefano Ermon. Adversarial constraint learning for structured prediction. *CoRR*, abs/1805.10561, 2018.

[20] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[21] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

[22] Evan Sheehan, Chenlin Meng, Matthew Tan, Burak Uzkent, Neal Jean, David Lobell, Marshall Burke, and Stefano Ermon. Predicting Economic Development using Geolocated Wikipedia Articles. *arXiv preprint arXiv:1905.01627*, 2019.

[23] Evan Sheehan, Burak Uzkent, Chenlin Meng, Zhongyi Tang, Marshall Burke, David Lobell, and Stefano Ermon. Learning to interpret satellite images using wikipedia. *arXiv preprint arXiv:1809.10236*, 2018.

[24] SpaceNet. Spacenet on amazon web services (aws). 'datasets.' the spacenet catalog, April 30, 2018.

[25] Russell Stewart and Stefano Ermon. Label-free supervision of neural networks with physics and domain knowledge. In *AAAI*, volume 1, pages 1–7, 2017.

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

[27] Burak Uzkent, Matthew J Hoffman, and Anthony Vodacek. Real-time vehicle tracking in aerial video using hyperspectral features. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 36–44, 2016.

[28] Burak Uzkent, Matthew J Hoffman, Anthony Vodacek, John P Kerekes, and Bin Chen. Feature matching and adaptive prediction models in an object tracking dddas. *Procedia Computer Science*, 18:1939–1948, 2013.

[29] Burak Uzkent, Aneesh Rangnekar, and Matthew Hoffman. Aerial vehicle tracking by adaptive fusion of hyperspectral likelihood maps. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 39–48, 2017.

[30] Burak Uzkent, Aneesh Rangnekar, and Matthew J Hoffman. Tracking in aerial hyperspectral videos using deep kernelized correlation filters. *IEEE Transactions on Geoscience and Remote Sensing*, (99):1–13, 2018.

[31] Liwei Wang, Yin Li, Jing Huang, and Svetlana Lazebnik. Learning two-branch neural networks for image-text matching tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018.

[32] Wikipedia. Wikipedia, the free encyclopedia, 2018.