What is it like down there?: generating dense ground-level views and image features from overhead imagery using conditional generative adversarial networks

Conference Paper · November 2018

DOI: 10.1145/3274895.3274969

CITATIONS
4

READS
47

3 authors:

Xueqing Deng
University of California, Merced
11 PUBLICATIONS **32** CITATIONS
SEE PROFILE

Yi Zhu
University of California, Merced
36 PUBLICATIONS **378** CITATIONS
SEE PROFILE

Shawn Newsam
University of California, Merced
75 PUBLICATIONS **1,987** CITATIONS
SEE PROFILE

What Is It Like Down There? Generating Dense Ground-Level Views and Image Features From Overhead Imagery Using Conditional Generative Adversarial Networks

Xueqing Deng
University of California, Merced
xdeng7@ucmerced.edu

Yi Zhu
University of California, Merced
yzhu25@ucmerced.edu

Shawn Newsam
University of California, Merced
snewsam@ucmerced.edu

ABSTRACT

This paper investigates conditional generative adversarial networks (cGANs) to overcome a fundamental limitation of using geotagged media for geographic discovery, namely its sparse and uneven spatial distribution. We train a cGAN to generate ground-level views of a location given overhead imagery. We show the "fake" ground-level images are natural looking and are structurally similar to the real images. More significantly, we show the generated images are representative of the locations and that the representations learned by the cGANs are informative. In particular, we show that dense feature maps generated using our framework are more effective for land-cover classification than approaches which spatially interpolate features extracted from sparse ground-level images. To our knowledge, ours is the first work to use cGANs to generate ground-level views given overhead imagery in order to explore the benefits of the learned representations.

CCS CONCEPTS

• **Information systems** → *Geographic information systems*; • **Computing methodologies** → *Image representations*; *Neural networks*;

KEYWORDS

Computer vision, generative adversarial networks, land-cover classification, geotagged social media

ACM Reference Format:
Xueqing Deng, Yi Zhu, and Shawn Newsam. 2018. What Is It Like Down There? Generating Dense Ground-Level Views and Image Features From Overhead Imagery Using Conditional Generative Adversarial Networks. In *Proceedings of ACM conference (SIGSPATIAL'18).* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3274895.3274969

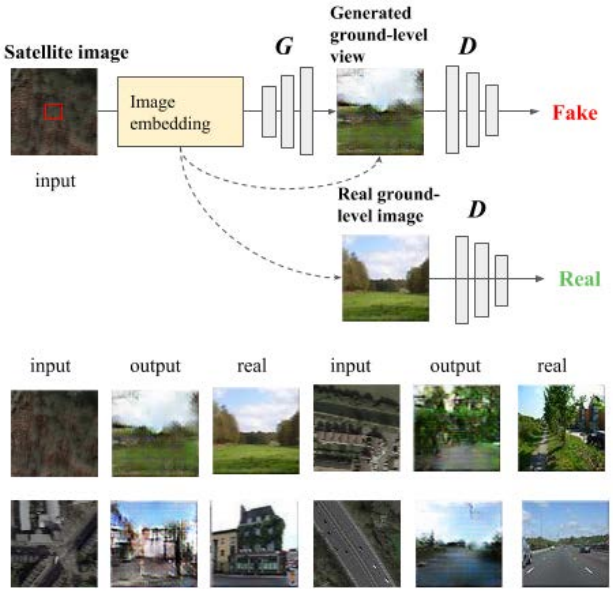

Figure 1: Overview of our work and selected results. Top: Proposed conditional generative adversarial network consisting of a generator which produces ground-level views given overhead imagery, and a discriminator which helps with training the generator as well as learning useful representations. Bottom: Select overhead image patches, the ground-level views generated by our our framework, and the real ground-level images.

1 INTRODUCTION

Mapping geographic phenomena on the surface of the Earth is an important scientific problem. The widespread availability of geotagged social media has enabled novel approaches to geographic discovery. In particular, "proximate sensing" [18], which uses ground-level images and videos available at sharing sites like Flickr and YouTube, provides a different perspective from remote sensing, one that can see inside buildings and detect phenomena not observable from above. Proximate sensing has been applied to map land use classes [38, 40], public sentiment [41], human activity [39], air pollution [19], and natural events [32], among other things. However, a fundamental limitation to using geotagged social media for mapping is its sparse and uneven spatial distribution. Unlike overhead imagery, it generally does not provide dense or uniform coverage.

This limitation restricts the kinds of maps that can be generated from geotagged social media. It also presents a challenge *when this data is fused with overhead imagery*. There has been great success on fusing overhead imagery with ground-level social media for problems such as land use classification [12, 21]. However, the maps produced by these approaches are at coarser spatial scales than the overhead imagery since the social media data is not available everywhere and therefore must be aggregated, say at the building or

SIGSPATIAL'18, November 2018, Seattle, Washington USA

ACM ISBN 978-1-4503-5889-7/18/11…$15.00
https://doi.org/10.1145/3274895.3274969

parcel level for land use classification. There has been less success on fusing the overhead imagery and ground-level social media at the resolution of the imagery. This is critical for producing finer scale, uniform maps as well as for areas where the aggregation units are not known a priori.

To our knowledge, Workman et al. [33] were the first to fuse overhead and ground-level imagery at the spatial resolution of the overhead imagery. They combine satellite and Google Street View (GSV) imagery to predict land use, building age, etc. They overcome the nonuniform spatial distribution of the GSV imagery by using Gaussian kernels to spatially interpolate features extracted from individual GSV images to match the resolution and coverage of the satellite imagery. However, as we showed in our previous work [5], such an approach is problematic in that it fails to infer the discontinuous spatial distribution of the ground-level image features. This motivates our novel work in this paper on more accurately estimating the spatial distribution of the *ground-level image features*. We do this by asking the question of what the ground-level view looks like on a dense spatial grid in order to derive the features. While this might seem to be an intractable problem, we believe that the recently proposed conditional generative adversarial networks provide an interesting solution.

Generative adversarial networks (GANs) have shown remarkable success in generating "fake" images that nonetheless look realistic. The key is learning the distribution of real-looking images in the space of all possible images. GANs accomplish this implicitly through a two player game in which a generator, given random noise as input, learns to generate images which a discriminator cannot tell apart from real images. Once trained, the generator can be used to produce novel images given random noise as input.

Our problem is a bit different, though. We do not want to simply generate ground-level views that look realistic (or, more accurately, whose features are from the distribution of real images). We also want to know how these ground-level views vary spatially. We thus turn to conditional GANs (cGANs) in which the generator and discriminator are conditioned on some additional information. This auxiliary information can be a simple class label, for example when generating real-looking images of the handwritten digits 0-9, or more complex data, possibly even in a different modality [24]. cGANs have been used to generate photo-realistic pictures from text descriptions [36] and to transfer styles between different visual domains, such as rendering a photograph as a painting or a satellite image as a map [15]. We perform a novel investigation into using cGANs to generate ground-level views conditioned on overhead imagery in order to produce dense feature maps.

The contributions of our work are as follows: (1) We propose a novel cGAN for generating ground-level views given overhead imagery. (2) We explore different representations/embeddings of the overhead imagery including image patches and convolutional neural network (CNN) features. (3) We demonstrate that our cGAN learns informative features that can be used, for example, to perform land-cover classification. (4) Finally, we compare the dense feature maps produced by our framework to those produced through interpolation in the context of land-cover mapping.

Our paper is organized as follows. Section 2 presents related work and Section 3 provides the technical details of our framework. Section 4 describes the datasets, provides visualizations of the generated ground-level views, and presents quantitative results of the land-cover classification. Section 5 concludes.

2 RELATED WORK

2.1 Conditional Generative Adversarial Nets

In the last few years, GANs have been explored for various computer vision problems and have shown great promise for generating detailed images [10] compared to previous generative methods which resulted in smoothed images. They are still difficult to work with, however, and instability in training makes it challenging to produce high-resolution or high-quality images. A number of techniques have therefore been proposed [1, 2, 6, 9, 23, 25, 26, 30] to stabilize the training as well as improve the results.

A number of interesting applications have been studied, particularly in the conditional setting, and include semantic face completion, which aims to recover a masked face [20, 34], image style transfer [3, 15, 37], face rotation [13, 31], pose estimation [4, 22], face generation [20], super resolution [17], semantic image generation [7] and so on.

Pix2Pix [15] performs image translation by processing the input images using an auto-encoder or a U-Net for the generator. A PatchGAN architecture is used for the discriminator. The skip connections in the U-Net, in particular, allow the generator to pass and thus preserve low-level characteristics of the input image, such as layout and shape, to the output image. Our problem is different though. There is no direct structural similarity between the overhead and the ground-level images as shown in Figure 1. While we expect the overhead image to be informative about what the ground-level view looks like, ours is not a style transfer problem and so the Pix2Pix framework is not appropriate.

StackGAN [36] proposes a two-stage GAN to transform text descriptions into photo-realistic images. Text descriptions are extracted by a pretrained encoder and then text embeddings are produced with Gaussian conditioning variables from the description in order to mitigate discontinuities which can cause instability during training. While this is not an image translation task, we take inspiration from it and investigate methods to extract embeddings of our overhead imagery.

2.2 Representation Learning of GANs

While GANs can amaze due to how realistic the generated images look, their real power often lies in the learned representations and the application of these representations to image analysis problems such as classification. Importantly, these representations are learned in an unsupervised manner since the training data need only consist of real images. For example, Raford et al. [26] propose a deep convolutional GAN architecture (DCGAN) to not only generate photo-realistic images but also extract features for classification. A DCGAN model trained on ImageNet-1k dataset[1] is used to extract features from the CIFAR-10 dataset[2]. An SVM is then used to perform classification with respect to the CIFAR-10 classes. The DCGAN features achieve just 4% lower accuracy than an Exemplar CNN model [8] which is trained directly on the CIFAR-10 dataset.

[1] http://www.image-net.org/
[2] https://www.cs.toronto.edu/ kriz/cifar.html

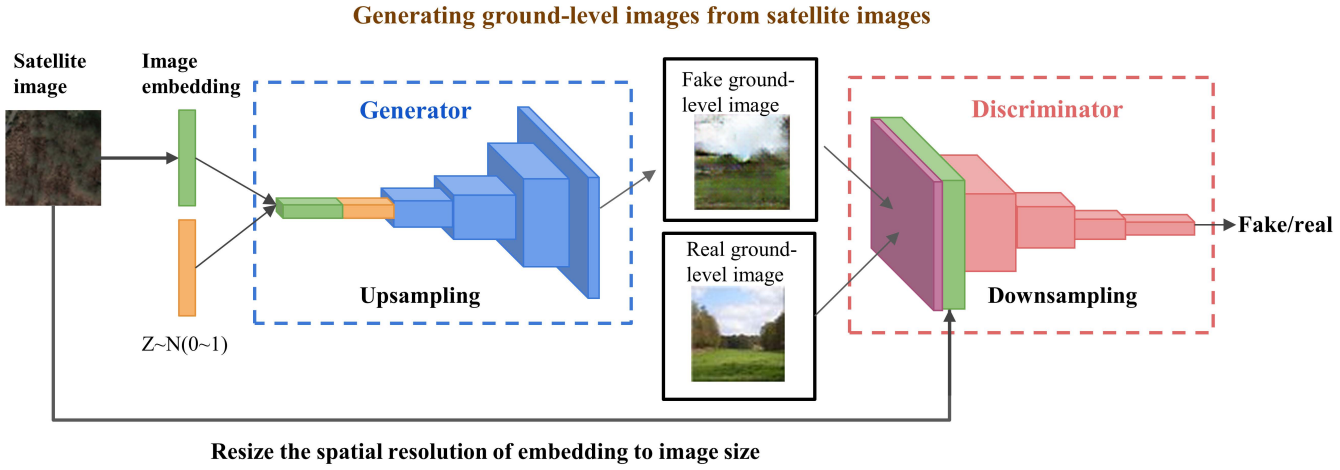

Figure 2: Network architecture: The generator takes as input an overhead image patch encoded to a vector and concatenated with a random vector. Its output is a realistic-looking but "fake" ground-level image. The discriminator tries to tell the difference between real and fake images. These two components play a competitive game during training.

This demonstrates the ability of GANs to learn useful features in an unsupervised manner. In our experiments below, we investigate the ability of GANs to learn useful features for land-cover classification.

2.3 Producing Dense Spatial Feature Maps

As already mentioned, a challenge to using geotagged social media for geographic discovery is its sparse and nonuniform distribution. Researchers have therefore investigated methods to spatially interpolate the *features* extracted from, for example, ground-level images to produce dense feature maps. This is the approach taken by Workman et al. in [33] to fuse VGG-16 features extracted from Google Street View images with high-resolution satellite imagery for dense land-use classification. However, as we showed in previous work [5], this interpolate-then-classify approach assumes that the ground-level features vary smoothly spatially which is often not the case. Instead, in this paper, we propose a novel method for generating dense ground-level feature maps by training a cGAN to generate fake ground-level images given overhead imagery and then using the learned representations as the features. The cGAN in a sense learns to predict "What is it like down there?" given overhead imagery.

2.4 Overhead to Ground-Level Cross-View Image Synthesis

We note there has been some prior and concurrent work on overhead to ground-level cross-view image synthesis. The prior work of Zhai et al. [35] learns to extract semantically meaningful features from overhead imagery by performing cross-view supervised training comparing the semantic segmentation of co-located Google Street View panoramas with segmentations generated from the overhead images that are transformed to ground-level view. The method is shown to be useful for weakly supervised overhead image segmentation, as well as a pre-training step for fully supervised segmentation; and for geo-locating and geo-orienting ground-level images. They do use the learned features to synthesize ground-level panoramas using a GAN-like architecture but these images are only visualized and not used for further analysis. And, only ground-level Street View images are considered. Street View images share significantly more structural similarity with co-located overhead images than our ground-level images which can be from any location, not just along streets.

Concurrent work by Regmi and Borji [28, 29] uses cGANs to synthesize ground-level Google Street View images using overhead imagery. However, the work is again limited to Street View images which share more structural similarity with overhead imagery. And, the goal is to produce visually high-quality ground-level images and not to use the synthesized images or their features for further geographic analysis.

3 METHODS

3.1 Overview of Our Methods

Our work has two goals. First, to generate natural-looking ground-level views given overhead imagery, and, second, to explore the learned representations for dense land-cover classification. Section 3.2 briefly introduces GANs and conditional GANs, and then describes our framework for generating ground-level views given overhead imagery. Section 3.3 describes the different embeddings we explore to input the overhead imagery to the cGAN as well as the objective function we use to train the cGAN. Section 3.4 provides the network architecture and implementation details. Finally, Section 3.5 describes how we access the learned representations by making modifications to the output layer of the discriminator.

3.2 GANs and cGANs

GANs [10] consist of two components, a generator and a discriminator. As shown in Figure 2, the generator G generates realistic looking but fake images by upsampling vectors of random noise. The discriminator's goal is to distinguish between real and fake

Table 1: Network architecture

(a) Generator

Name	kernel	channel in/out	In res	Out res	Input
deconv1	4×4	nef+100 /1024	1×1	4×4	embedding+ random noise
deconv1_bn	Batchnorm				deconv1
deconv2	4×4	1024/512	4×4	8×8	deconv1_bn
deconv2_bn	Batchnorm				deconv2
deconv3	4×4	512/256	8×8	16×16	deconv2_bn
deconv3_bn	Batchnorm				deconv3
deconv4	4×4	256/128	16×16	32×32	deconv3_bn
decon4_bn	Batchnorm				deconv4
deconv5	4×4	128/3	32×32	64×64	deconv4_bn

(b) Discriminator

Name	kernel	channel in/out	In res	Out res	Input
conv1_1	4×4	3/64	64×64	32×32	image
conv1_bn1	Batchnorm				conv1_1
conv1_2	4×4	nef/64	64×64	32×32	embedding
conv1_bn2	Batchnorm				conv1_2
conv2	4×4	128/256	32×32	16×16	conv1_bn1+ conv1_bn2
conv2_bn	Batchnorm				conv2
conv3	4×4	125/512	16×16	8×8	conv2_bn
conv3_bn	Batchnorm				conv3
conv4	4×4	512/1024	8×8	4×4	conv3_bn
con4_bn	Batchnorm				conv4
conv5	4×4	1024/1	4×4	1×1	conv4_bn

images through downsampling. GANs learn generative models through adversarial training. That is, G tries to fool D. G and D are trained simultaneously. G is optimized to reproduce the true data distribution by generating images that are difficult to distinguish from real images by D. Meanwhile, D is optimized to differentiate fake images generated by G from real images. Overall, the training procedure is similar to a two-player min-max game with the following objective function

$$\min_G \max_D V(D,G) = \mathbb{E}_{x \backsim p_{data}(x)}[\log D(x)] + \mathbb{E}_{z \backsim p_{data}(z)}[1 - \log D(G(z))] \quad (1)$$

where z is the random noise vector and x is the real image. Once trained, G can be used to generate new, unseen images given random vectors as input. In practice, rather than train G to minimize $1 - \log D(G(z))$, we train G to maximize $\log D(G(z))$, as demonstrated in [10].

GANs, as a generative model, learn the distribution of the entire training dataset. Conditional GANs were therefore introduced to learn distributions conditioned on some auxiliary information. For example, a GAN can be trained to generate realistic-looking images of hand written digits from the MNIST dataset. However, if we want images of a specific digit, a "1" for example, the generative model needs to be conditioned on this information. In cGANs, the auxiliary information y is incorporated through hidden layers separately in both the generator and discriminator. y can take many forms, such as class labels [24], text embeddings for generating images from text [27, 36], and images for image translation [15, 37]. The cGAN objective function becomes

$$\min_G \max_D V(D,G) = \mathbb{E}_{x \backsim p_{data}(x)}[\log D(x|y)] + \mathbb{E}_{z \backsim p_{data}(z)}[1 - \log D(G(z|y))]. \quad (2)$$

In the context of our problem, we use overhead imagery as the auxiliary information to generate natural-looking ground-level views. However, ours is not an image translation problem since there is no direct structural similarity between the overhead images and the ground-level views. We therefore consider embeddings of the overhead imagery.

3.3 Overhead Image Embedding

The overarching premise of our novel framework is that overhead imagery contains information about what things look like on the ground. It is not obvious, however, how to represent this information so that it can be input to a cGAN to generate ground-level views. We know that simply feeding a 2D overhead image patch to the cGAN and then performing image translation does not make sense due to the lack of structural similarity between the overhead and ground-level images. We are also constrained by the fact that the input to GANs/cGANs cannot be too large otherwise the learning will become unstable.

Below, we describe three types of embeddings, each of which produces a 1D vector which is concatenated with the noise vector and then input to the generator. As is standard in cGANs, the overhead image information is also provided to the discriminator. See Figure 2 for details.

The objective function for training our cGAN to generate ground-level views from overhead imagery becomes

$$L_D = \mathbb{E}_{(x,\varphi(I_s)) \backsim p_{data}(I_g,\varphi(I_s))} + \mathbb{E}_{z \backsim p_z, \varphi(I_s) \backsim p_{data}}[1 - \log D(G(z,\varphi(I_s)),\varphi(I_s))] \quad (3)$$

$$L_G = \mathbb{E}_{z \backsim p_z, \varphi(I_s) \backsim p_{data}}[1 - \log D(G(z,\varphi(I_s)),\varphi(I_s))] \quad (4)$$

where I_s and I_g refer to satellite image and ground-level images respectively, and $\varphi(I)$ is the overhead image embedding.

Grayscale image patch embedding Our baseline embedding extracts a 10×10 pixel patch from the overhead imagery centered on where we want to generate the ground-level view, computes the grayscale pixel values as the average RGB values, reshapes the patch into a vector, and normalizes the components to range between -1 and 1. This results in a 100D vector.

HSV image patch embedding In order to investigate whether the color information in the overhead imagery is useful for our problem, we convert the 10×10 patches from RGB to HSV colorspaces and then form a normalized, 300D vector.

CNN image feature embedding The above embeddings are straightforward encodings of the overhead image patches. We also consider using a pre-trained VGG-16 model to encode the image patches as CNN features. This results in 1024D vectors which we reduce to 25D through dimensionality reduction. The motivation here is that CNNs have been shown to produce effective image encodings for a range of computer vision tasks.

3.4 Network Architecture and Implementation Details

GANs architectures can consist of either multi-layer perceptrons or convolutional networks. We adopt CNNs as they have been shown to be more stable during training [26]. Our CNNs consist of convolutional (conv) layers, followed by batchnorm [14] and leakyReLU steps, and so are referred to as a conv-batchnorm-leakyReLU architecture. Strided convolutions are used to increase or decrease the image resolution instead of maxpooling. Table 1 provides the details of our generator and discriminator networks. In Table 1, (a) and (b) are the network architectures for generator and discriminator individually, where *deconv* denotes a transposed convolution layer, *conv* denotes a convolution layer, *bn* denotes batchnorm and nef denotes the number of dimensions of the embedding function output.

The input to our generator is the concatenation of a 100D random noise vector and an nef dimensional embedding of the overhead image patch. We concatenate these vectors before inputting them to the conv-batchnorm-leakyReLU architecture similar to the work on text-to-image generation in [36]. This is different from the work in [24] in which the two vectors are separately fed through convolutional layers and later concatenated. We find that concatenating first makes the training more stable. The output of our generator is a 64×64 pixel RGB ground-level image.

Our discriminator follows the same conv-batchnorm-leakyReLU architecture but the last layer is a sigmoid activation function which outputs a binary value corresponding to real or fake. The discriminator takes as input a ground-level image, real or fake, and the auxiliary information in the form of the overhead image patch. In the case of a real image, the overhead image patch is the actual overhead view of where the image is located. In the case of a fake image, the overhead image patch is what was used by the generator to produce the image. The output of the discriminator is its belief of whether the ground-level image is real or fake. Two different losses are used to train the discriminator depending on whether the input ground-level image is real or fake.

We implement our deep learning framework using PYTORCH[3] and ADAM [16] is used as our optimizer. The initial learning rate is set to 0.0002. We train our cGAN for 400 epochs with a batch size of 128 on one NVIDIA GTX 980Ti GPU.

[3]https://pytorch.org

3.5 Generating Dense Feature Maps For Land-Cover Classification

Our goal is not just to see how well our generator can produce ground-level views given overhead imagery, but also to investigate whether our entire cGAN, the generator and discriminator, can learn novel representations for tasks such as classification. As an example application, we explore whether these representations are effective for land-cover classification. This would then allow us to generate dense ground-level feature maps given only overhead imagery.

We modify our cGAN to output features as follows. Following [26], we remove the last, sigmoid layer of our discriminator and add an average pooling layer. Our cGAN then outputs a 1024D feature vector given an overhead image patch. This feature can then be used to perform analysis at the spatial resolution of the overhead imagery (by using overlapping patches), such as per pixel classification, image segmentation, etc.

4 EXPERIMENTS

4.1 Datasets and Land-Cover Classes

Our framework requires co-located ground-level and overhead imagery. We download ground-level images with known location from the Geograph API[4]. We download georeferenced overhead imagery from the Google Map Static API[5].

For land-cover classification, we use the ground-truth land-cover map LCM2015[6] to construct our training and test datasets. This map provides land-cover classes for the entire United Kingdom on a 1km grid. We group these classes into two super-classes, urban and rural, and limit our study to a 71km×71km region containing London. The Geograph images are labeled as urban or rural based on the LCM2015 label of the 1km×1km grid cell from which they are downloaded. We realize this label propagation likely results in some noisy labels in our dataset.

Figure 3 shows that the Geograph images corresponding to our two land-cover classes are visually very different.

4.2 Preprocessing Data

We resize our real ground-level images to measure 64×64 pixels to match the size of the images produced by our generator. This allows us to compare classification results of using the real images with the results of using the fake images.

4.3 Generating Ground-Level Views Given Overhead Imagery

We trained our cGAN using 4,000 Geograph images and co-located overhead image patches split evenly between urban and rural locations. We then used our generator to produce ground-level views given overhead image patches at other locations. Figure 4 shows the overhead image patches and generated ground-level views corresponding to the three different overhead image embeddings. Also shown are the true ground-level images.

[4]http://www.geograph.org.uk/
[5]https://developers.google.com/maps/documentation/maps-static/intro
[6]https://eip.ceh.ac.uk/lcm/lcmdata

Figure 3: Examples of Geograph images for (a) urban locations and (b) rural locations.

Table 2: Land-cover classification accuracy with features

Classifier	Features		Dimension	Accuracy
	Type	Name		
SVM	Embedding from Google satellite images	Grayscale image patches	100	93.17
		HSV image patches	300	**93.47**
		VGG-extracted image features	25	93.19
	cGAN generated image features conditioned on different embeddings	Grayscale image patches	1024	**82.33**
		HSV image patches feature	1024	74.33
		VGG-extracted image features	1024	61.82
	cGAN generated image features conditioned on different embeddings + embeddings	Grayscale image patches + image patches	1024+100	**86.34**
		HSV image patches + image patches	1024+300	75.13
		VGG-extracted image features + image features	1024+25	65.94
ResNet-34	Ground-level images	Interpolated CNN features	512	58.5

Despite the difficulty of our task, the results in Figure 4 demonstrate that our proposed cGAN framework is able to produce surprisingly reasonable ground-level views given the overhead image patches. The results of the grayscale embedding are the most realistic looking and generally align well with the real images. This is particularly true for the rural scenes with lots of grass, trees and sky. Urban scenes are more complex with detailed objects with crisp boundaries and the results look less realistic. However, similar to the real images, *the generated ground-level views corresponding to the two land-cover classes are visually very distinct*, at least for the grayscale embedding.

The results of the HSV embedding are not as realistic looking as that of the grayscale embedding. This is somewhat surprising since it seems the additional color information would be informative. The 300D vector embeddings might simply be too large for training the cGAN.

The results of the CNN feature embedding are poor. The images are not realistic looking and there is not much distinction between the two land-cover classes. We will investigate this further in future work as it seems the CNN features should be able to encode the overhead image information.

In summary, while it is unlikely the generated ground-level views would be mistaken for real images, the results of the grayscale and HSV embeddings are very representative of what things look like on the ground. We believe this is an impressive outcome given that these images are generated solely using 10×10 pixel overhead image patches.

4.4 Generating Ground-Level Image Features Given Overhead Imagery

We now investigate whether our trained cGAN has learned representations useful for other tasks such as classification. We compare land-cover classification using our learned representation with interpolating between real ground-level images.

We download Geograph images and co-located overhead imagery corresponding to 20,000 locations. The land-cover classes of these locations are known from the LCM2015 ground truth map. These locations are split into 16,000 training locations and 4,000 test locations.

We first make sure that the overhead image patches themselves are representative of the land-cover classes. We train SVM classifiers (with RBF kernels) using the different embeddings of the overhead image patches as input (again, we know the classes of these patches). We then apply the SVMs to the overhead image patches corresponding to the test locations. The results are shown at the top of Table 2. All three classifiers achieve over 93% accuracy on the test set which indicates the overhead image patches are representative of the two classes and are thus suitable candidates as auxiliary information for our cGANs.

Figure 4: Selected images generated by our proposed cGAN on (a) rural locations and (b) urban locations. The columns for each group from left to right are the input overhead images (extracted image patches are indicated by the red boxes), the cGAN generated images generated using HSV, CNN, and grayscale encoded patches, and the real ground-level images.

We now evaluate our learned representation. For each of the 20,000 locations, we generate a 1024D feature by applying our modified cGAN (see Section 3.5) to the overhead image patch for that location. We then train and test an SVM classifier. This is done separately for the three different overhead image embeddings. The classification results are shown in Table 2. The grayscale embedding performs the best, achieving an accuracy of 82.33%, demonstrating that our cGAN has learned a useful representation. For context, learning a ResNet-34 [11] classifier on the 16,000 real ground-level training images and then applying it to the 4,000 real ground-level test images results in 88.2% accuracy. The performance of our learned representation is not far off this gold standard.

Table 3: Land-cover classification with dense feature map

Name	Accuracy
Geograph images	69.07
cGAN generated features	**73.14**
Interpolated features	65.86

The classification results of the HSV and CNN feature embeddings are 72.12% and 61.82% respectfully. Again, the grayscale embedding seems best.

We now evaluate the performance of interpolating between real ground-level images. We divide the ResNet-34 classifier learned from the 16,000 real ground-level training images into a feature extractor and a classifier. The feature extractor is applied to the 16,000 training images to derive a 512D feature for each of the 16,000 locations. The 512D features at the 4,000 test locations are derived by spatially interpolating the features at the 16,000 training locations using a Gaussian kernel. The classifier is then applied to the interpolated features. As shown in Table 2, this results in an accuracy of only 58.5%, much worse than our proposed cGAN framework which is able to achieve 82.33%.

Finally, for completeness, we concatenate the cGAN learned representations and the overhead image embeddings, and train and evaluate an SVM classifier. As shown in Table 2, this improves upon the performance of only using the cGAN representations.

4.5 Creating Dense Feature Maps

In our final experiment, we use our cGAN framework to produce a dense ground-level feature map given overhead imagery. This feature map is then used to generate a land-cover map. We compare this map with ones produced by classifying densely-sampled ground-level images as well by an interpolate-then-classify [5, 33] approach applied to sparsely-sampled images.

The left image in Figure 5 shows the ground truth map based on LCM2015. Again, this is a 71×71 gridded region in which each cell corresponds to a 1km×1km square.

To produce a land-cover map based on densely-sampled ground-level images, we download 10 Geograph images for each cell, keeping track of the locations. This results in $71 \times 71 \times 10 = 50,410$ images in total. The ResNet-34 classifier trained earlier is applied to the 10 images in each grid cell and the majority label is assigned to the cell. This results in the second map from the left in Figure 5.

To produce a land-cover map based on our cGAN framework, we use our modified cGAN to extract features from overhead patches at the locations of the 50,410 images. We then apply the SVM classifier trained earlier to the features for the 10 locations in each grid cell and again assign the majority label to the cell. This results in the third map from the left in Figure 5.

Finally, to produce a land-cover map using an interpolate-then-classify approach, we use the ResNet-34 feature extractor to extract features from just 836 of the 50,410 ground-level images. Spatial interpolation is then used to estimate the features at the remaining locations and the ResNet-34 classifier is applied. Implementation details can be found in [5]. The majority label is again used to label the grid cells. This results in the map on the right in Figure 5.

Qualitatively, the map produced using the proposed cGAN framework is more similar to the ground truth than those generated using the densely-sampled ground-level images and the interpolate-then-classify approach. The ground-level images are quite heterogeneous and can vary quite a bit within a 1km×1km region. A cell will be labeled incorrectly if it happens that the majority of the images depict a class other than the ground truth. And, as expected, the interpolate-then-classify approach results in a smoothed map. Interpolation is not able to detect islands of one class that are surrounded by another class.

Table 3 shows a quantitative comparison between the ground truth and predicted maps. Overall accuracy is computed as the percentage of grid cells for which the ground truth and predicted maps agree. The proposed cGAN framework is shown to perform the best.

4.6 Limitations of the Generated Views

We investigate whether the generated images themselves, as opposed to the learned representations, are useful for classification. We generate 20,000 fake images corresponding to the 20,000 locations used in Section 4.4 using overhead image patches. We then learn a ResNet-34 classifier using the 16,000 fake images corresponding to the training locations and apply it to the 4,000 fake images corresponding to the test locations. The classification accuracies corresponding to the three overhead image embeddings are shown in Table 4. Also shown on the top row is the accuracy of the ResNet-34 classifier learned and applied to the real ground-level images. The fake images are seen not to be as effective for classification as the real images or the learned representations (see Table 2). This is likely a result of our generated images lacking the details that the ResNet-34 classifier is able to exploit in real images.

5 CONCLUSIONS AND FUTURE WORK

We investigate cGANs for generating ground-level views and the corresponding image features given overhead imagery. The generated ground-level images look natural, although, as expected, they lack the details of real images. They do capture the visual distinction between urban and rural scenes. We show the learned representations are effective as image features for land-cover classification. In particular, the representations are almost as effective at classifying locations into urban and rural classes as the real ground-level images. We use the cGANs to generate dense feature maps. These feature maps are more effective for producing land-cover maps than dense samples of ground-level images. Our proposed method is not limited to land-cover classification. It provides a framework to create dense feature maps for other applications.

This represents our preliminary work on this problem. We plan to develop cGANs that can generate more detailed ground-level views that can be used directly for image classification, etc. The training of the cGANs is still very unstable. We will therefore also investigate other techniques and architectures to make the training of cGANS more stable for our particular problem.

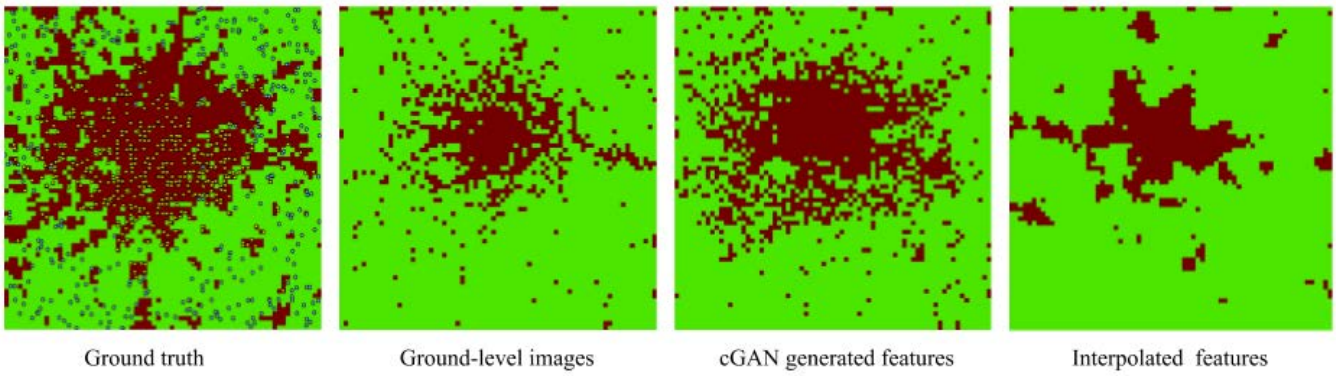

Figure 5: Ground truth and predicted land-cover maps. From left to right: ground truth map; map generated using ResNet-34 applied to a dense sampling of ground-level images; map generated using cGAN generated features; and map generated using interpolated features. (brown: urban, green: rural) The dots in the ground truth represent the locations of the sparse ground-level images used in the interpolation (yellow: urban, blue: rural)

Table 4: Land-cover classification accuracy using ground-level images

Classifier	Images	Accuracy
ResNet-34	Real ground-level images	88.2
	cGAN generated fake images with grayscale overhead image embedding	62.8
	cGAN generated fake images with HSV overhead image embedding	62.5
	cGAN generated fake images with CNN features overhead image embedding	61.8

6 ACKNOWLEDGMENTS

This work was funded in part by a National Science Foundation CAREER grant, #IIS-1150115. We gratefully acknowledge the support of NVIDIA Corporation through the donation of the GPU card used in this work

REFERENCES

[1] M. Arjovsky, S. Chintala, and L. Bottou. 2017. Wasserstein Generative Adversarial Networks. In *Proceedings of the 34th International Conference on Machine Learning*, Doina Precup and Yee Whye Teh (Eds.), Vol. 70. PMLR, International Convention Centre, Sydney, Australia, 214–223.

[2] D. Berthelot, T. Schumm, and L. Metz. 2017. BEGAN: Boundary Equilibrium Generative Adversarial Networks. *ArXiv e-prints* (March 2017). arXiv:cs.LG/1703.10717

[3] K. Bousmalis, N. Silberman, D. Dohan, D. Erhan, and D. Krishnan Krishnan. 2017. Unsupervised Pixel-Level Domain Adaptation with Generative Adversarial Networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 95–104.

[4] Y. Chen, C. Shen, X.-S. Wei, L. Liu, and J. Yang. 2017. Adversarial PoseNet: A Structure-aware Convolutional Network for Human Pose Estimation. *ArXiv e-prints* (April 2017). arXiv:cs.CV/1705.00389

[5] X. Deng, Y. Zhu, and S. Newsam. 2018. Spatial Morphing Kernel Regression for Feature Interpolation. In *25th IEEE International Conference on Image Processing (ICIP)*. 2182–2186.

[6] E. Denton, S. Chintala, A. Szlam, and R. Fergus. 2015. Deep Generative Image Models Using a Laplacian Pyramid of Adversarial Networks. In *Proceedings of 28th International Conference on Neural Information Processing Systems*. 1486–1494.

[7] H. Dong, S. Yu, C. Wu, and Y. Guo. 2017. Semantic Image Synthesis via Adversarial Learning. In *Proceedings of the IEEE International Conference on Computer Vision*. 5707–5715.

[8] A. Dosovitskiy, P. Fischer, J. Springenberg, M. Riedmiller, and T. Brox. 2016. Discriminative Unsupervised Feature Learning with Exemplar Convolutional Neural Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 38, 9 (2016), 1734–1747.

[9] I. Goodfellow. 2017. NIPS 2016 Tutorial: Generative Adversarial Networks. *ArXiv e-prints* (Dec. 2017). arXiv:cs.LG/1701.00160

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

[11] K. He, X. Zhang, S. Ren, and J. Sun. 2016. Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 770–778.

[12] T. Hu, J. Yang, X. Li, and P. Gong. 2016. Mapping Urban Land Use by Using Landsat Images and Open Social Data. *Remote Sensing* 8 (2016), 151.

[13] R. Huang, S. Zhang, T. Li, and R. He. 2017. Beyond Face Rotation: Global and Local Perception GAN for Photorealistic and Identity Preserving Frontal View Synthesis. In *Proceedings of the IEEE International Conference on Computer Vision*. 2458–2467.

[14] S. Ioffe and C. Szegedy. 2015. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In *32nd International Conference on Machine Learning (Proceedings of Machine Learning Research)*, Francis Bach and David Blei (Eds.), Vol. 37. PMLR, Lille, France, 448–456.

[15] P. Isola, J. Zhu, T. Zhou, and A. A Efros. 2017. Image-to-Image Translation with Conditional Adversarial Networks. In *Proceedings of the IEEE International Conference on Computer Vision*.

[16] D. P. Kingma and J. Ba. 2014. Adam: A Method for Stochastic Optimization. *CoRR* abs/1412.6980 (2014).

[17] C. Ledig, L. Theis, F. HuszÃąr, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi. 2017. Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

[18] D. Leung and S. Newsam. 2010. Proximate Sensing: Inferring What-Is-Where From Georeferenced Photo Collections. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2955–2962.

[19] Y. Li, J. Huang, and J. Luo. 2015. Using User Generated Online Photos to Estimate and Monitor Air Pollution in Major Cities. In *Proceedings of the 7th International Conference on Internet Multimedia Computing and Service*. 79.

[20] Y. Li, S. Liu, J. Yang, and M. Yang. 2017. Generative Face Completion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

[21] X. Liu, J. He, Y. Yao, J. Zhang, H. Liang, H. Wang, and Y. Hong. 2017. Classifying Urban Land Use by Integrating Remote Sensing and Social Media Data. *International Journal of Geographical Information Science* 31, 8 (2017), 1675–1696.

[22] L. Ma, X. Jia, Q. Sun, B. Schiele, T. Tuytelaars, and L. Van Gool. 2017. Pose Guided Person Image Generation. In *Advances in Neural Information Processing Systems (NIPS)*. 405–415.

[23] X. Mao, Q. Li, H. Xie, R. Y. K. Lau, Z. Wang, and S. P. Smolley. 2016. Least Squares Generative Adversarial Networks. *ArXiv e-prints* (Nov. 2016). arXiv:cs.CV/1611.04076

[24] M. Mirza and S. Osindero. 2014. Conditional Generative Adversarial Nets. *arXiv preprint arXiv:1411.1784* (2014).

[25] S. Mohamed and B. Lakshminarayanan. 2016. Learning in Implicit Generative Models. *ArXiv e-prints* (Oct. 2016). arXiv:stat.ML/1610.03483

[26] A. Radford, L. Metz, and S. Chintala. 2016. Unsupervised representation learning with deep convolutional generative adversarial networks. In *International Conference on Representation Learning (ICRL)*.

[27] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee. 2016. Generative Adversarial Text to Image Synthesis. In *Proceedings of the 33rd International Conference on Machine Learning (ICML) (Proceedings of Machine Learning Research)*, Maria Florina Balcan and Kilian Q. Weinberger (Eds.), Vol. 48. PMLR, New York, New York, USA, 1060–1069.

[28] K. Regmi and A. Borji. 2018. Cross-View Image Synthesis Using Conditional GANs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

[29] K. Regmi and A. Borji. 2018. Cross-view image synthesis using geometry-guided conditional GANs. *ArXiv e-prints* (Aug. 2018). arXiv:cs.CV/1808.05469

[30] T. Salimans, I. Goodfellow, W. Zaremba, V.i Cheung, A. Radford, and X. Chen. 2016. Improved Techniques for Training GANs. In *Advances in Neural Information Processing Systems (NIPS)*. 2234–2242.

[31] L. Tran, X. Yin, and X. Liu. 2017. Representation Learning by Rotating Your Faces. *ArXiv e-prints* (May 2017). arXiv:cs.CV/1705.11136

[32] J. Wang, M. Korayem, S. Blanco, and D. Crandall. 2016. Tracking Natural Events Through Social Media and Computer Vision. In *2016 ACM on Multimedia Conference*. 1097–1101.

[33] S. Workman, M. Zhai, D. J. Crandall, and N. Jacobs. 2017. A Unified Model for Near and Remote Sensing. In *Proceedings of the IEEE International Conference on Computer Vision*. 2707–2716.

[34] R. A. Yeh*, C. Chen*, T. Lim, A. G. Schwing, M. Hasegawa-Johnson, and M. N. Do. 2017. Semantic Image Inpainting with Deep Generative Models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. * equal contribution.

[35] M. Zhai, Z. Bessinger, S. Workman, and N. Jacobs. 2017. Predicting Ground-Level Scene Layout from Aerial Imagery. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

[36] H. Zhang, T. Xu, H. Li, S. Zhang, X. Huang, X. Wang, and D. Metaxas. 2017. Stackgan: Text to Photo-Realistic Image Synthesis with Stacked Generative Adversarial Networks. In *Proceedings of the IEEE International Conference on Computer Vision*. 5907–5915.

[37] J. Zhu, T. Park, P. Isola, and A. A Efros. 2017. Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks. In *Proceedings of the IEEE International Conference on Computer Vision*.

[38] Y. Zhu, X. Deng, and S. Newsam. 2018. Fine-Grained Land Use Classification at the City Scale Using Ground-Level Images. *ArXiv e-prints* (Feb. 2018). arXiv:cs.CV/1802.02668

[39] Y. Zhu, S. Liu, and S. Newsam. 2017. Large-Scale Mapping of Human Activity Using Geo-Tagged Videos. In *Proceedings of the 25th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. 68:1–68:4.

[40] Y. Zhu and S. Newsam. 2015. Land Use Classification Using Convolutional Neural Networks Applied to Ground-Level Images. In *Proceedings of the 23rd SIGSPATIAL International Conference on Advances in Geographic Information Systems*. 61.

[41] Y. Zhu and S. Newsam. 2016. Spatio-Temporal Sentiment Hotspot Detection Using Geotagged Photos. In *Proceedings of the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. 76.