Densely Labeling Large-Scale Satellite Images with Generative Adversarial Networks

Yupeng Yan, *Xiaohui Huang*, *Anand Rangarajan* and *Sanjay Ranka*
University of Florida, Gainesville, Florida, USA
{yanyp, xiaohuihuang}@ufl.edu, {anand, ranka}@cise.ufl.edu

***Abstract*—Building an efficient and accurate pixel-level labeling framework for large-scale and high-resolution satellite imagery is an important machine learning application in the remote sensing area. Due to the very limited amount of the ground-truth data, we employ a well-performing superpixel tessellation approach to segment the image into homogeneous regions and then use these irregular-shaped regions as the foundation for the dense labeling work. A deep model based on generative adversarial networks is trained to learn the discriminating features from the image data without requiring any additional labeled information. In the subsequent classification step, we adopt the discriminator of this unsupervised model as a feature extractor and train a fast and robust support vector machine to assign the pixel-level labels. In the experiments, we evaluate our framework in terms of the pixel-level classification accuracy on satellite imagery with different geographical types. The results show that our dense-labeling framework is very competitive compared to the state-of-the-art methods that heavily rely on prior knowledge or other large-scale annotated datasets.**

***Keywords*—Superpixel segmentation, unsupervised feature learning, generative adversarial networks, remote sensing**

I. Introduction

The rapid development and deployment of remote sensing instruments has resulted in huge corpora of large-scale satellite images with very high resolution (VHR). This generates an urgent need to efficiently and accurately understand these massive datasets. In recent years, we have seen many efforts made to develop an automated computational architecture for densely labeling such satellite imagery at the pixel level and providing the semantic interpretation of different areas, such as the human buildings, agricultural fields, primitive forests and bodies of water. Fig. 1 shows an example that contains the common types of terrain objects. The filled circles represent the data labeled by the remote sensing experts.

Although building a successful dense-labeling framework looks interesting and promising, we should point out the difficulties and challenges that we are facing, because the following issues may not be well addressed:

- First, we are often provided a very limited number of ground truth points (in terms of the pixel coordinates) that are labeled by the experts. Compared to the actual scale of the satellite images which consists of millions or a few billion pixels, the labeled data only occupies a small proportion and machine learning approaches that only work with a small number of labels is feasible.
- Second, we should realize that the expert-labeled point is more likely to represent several pixels or large homogeneous regions rather than a single "independent" pixel. Thus, the local region homogeneity needs to be considered and many of the pixel-based image processing approaches cannot deal with this information.
- Third, due to the lack of training samples, we are unable to employ the effective supervised learning approaches to obtain the discriminating representations from the image. Prior knowledge and experience still play a key role in extracting the high-level semantic features, which creates additional uncertainty regarding the rarely seen objects.

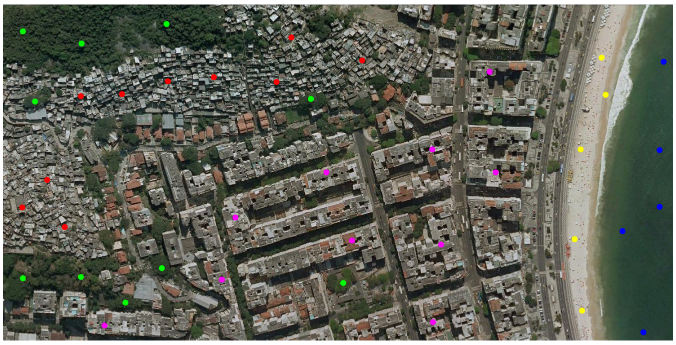


Fig. 1: An example of a large-scale VHR satellite image captured from Rio, Brazil. The solid circles show the expert-labeled points given in terms of the pixel coordinates. Different colors represent different labeling categories, such as slum (red), urban (magenta), forest (green), beach sand (yellow) and sea (blue).

Some of the previous works have addressed the first two issues above by employing a *superpixel* tessellation approach to segment the large-scale image into *coherent regions* that are used as the basic labeling unit. This processing step can significantly reduce the computational complexity in the overall labeling work and help to fill in the quantity gaps between the labeled and unlabeled data. Moreover, the segmentation is also able to better capture the coherent local structure of the terrain objects, so the classification quality can be improved on the pixels near the boundaries between the two different neighboring objects.

Furthermore, we notice that recent unsupervised learning approaches have made huge progress on the feature learning and selection tasks as the deep network structure becomes widely adopted. Generative adversarial networks (GAN) are the most promising methods to learn the deep representation since they do not require any labeled information. This advantage makes the GAN a great choice for the feature learning step in the dense-labeling framework, because we have a tremendous volume of unlabeled data (from the large-scale VHR images) to train a robust deep model without triggering the overfitting problem. In this way, we can address the third issue without relying on any additional knowledge or experience from the remote sensing field.

The main contribution of this paper is summarized in the following aspects:

- We effectively adopt superpixel segmentation to leverage the limited amount of expert-labeled points and improve the computational efficiency of our framework.

This work is partially supported by NSF IIS 1743050.

- This is the first successful attempt at using GANs in the dense labeling framework for large-scale VHR satellite images. Our approach is more accurate than many existing algorithms and competitive to the state-of-the-art deep learning models where we have to perform transfer learning from other well-annotated datasets.
- To the best of our knowledge, this is the first time the GAN scheme has been successfully integrated with the transfer learning method. We use this approach to further improve the recognition ability and achieve good results on our VHR imagery.

We organize the rest of the paper as follows. In Section II, the related works are briefly introduced. Our dense-labeling framework is described in Section III. We report out experimental results in Section IV and present the conclusions in Section V.

II. Related Work

Many effective low-level descriptors and middle-level features, such as color histogram, normalized difference vegetation index (NDVI), scale-invariant feature transform (SIFT), bag of visual words (BoVW) and semantic-spatial matching (SSM) [1], play an important role in the recent dense labeling framework for large-scale images. Vatsavai [2] proposed a multiple instance approach based on the low-level remote sensing features. The neighboring pixels were modeled with a Gaussian distribution in order to capture the complex spatial patterns. Sethi *et al.* [3], [4], [5] also took the advantages of these descriptors by concatenating them in to a long vector and then use it as the representational features. They assigned a set of reasonable weights to the descriptors and made their framework perform well on the global-scale application.

Furthermore, the deep learning algorithms such as the convolutional neural networks (CNN) have been developed very fast in recent years. They are able to learn the useful high-level semantic features from the images and achieve the-state-of-the-art classification performance in many remote sensing tasks. It overcomes the drawback of the handcrafted feature selection process, which was completely relying on the human knowledge in the area of expertise. For example, [6], [7] have demonstrated the great advantages of CNN for the hyperspectral image classification and semantic scene recognition respectively.

On the other hand, we should note that all these CNN-based methods need a huge amount of labeled data to train a robust deep model. It imposes an impractical requirement for designing a dense labeling framework because in most situations, we are only available to a small fraction or even very limited number of the labeled samples. To this end, instead of constructing the deep learning model from scratch, *et al.* [8] borrowed the successful CNN models that are pre-trained from a huge everyday object recognition dataset. They removed the last fully-connected layer and treated the remaining network as a fixed feature extractor. This transfer learning method retained the strong generalization ability of the deep features and could perform even better by concatenating the features from multiple CNN models.

Moreover, the unsupervised deep learning algorithms are seen as a more promising method since they can learn the maximum effectiveness to reconstruct the original data without using any labeled information. Xu *et al.* [9] presented a stacked sparse autoencoder framework to learn the high-level features of the input raw data. Better representation can be obtained comparing to the conventional feature reduction methods like the principal component analysis. The generative adversarial networks (GAN) [10] is another excellent choice to learn the unsupervised features, because its generator can provide large quantities of the fake data from the random vectors, and thus we only need to provide the real data from the real satellite imagery to train its discriminator. Radford *et al.* [11] introduced the deep convolutional GANs (DCGAN) to bridge the gap to the success of supervised learning by utilizing the CNN architectures. Their trained discriminators showed competitive performance on the bedroom, face and natural object datasets. Lin *et al.* [12] presented a multi-layer feature-matching GAN (MARTA-GAN) to avoid the checkerboard artifacts in DCGAN and generate the fake (or synthetic) images with higher resolution. A multi-feature layer was added to the architecture of DCGAN so the perceptual loss could be combined with their feature matching loss.

In this work, we design our dense labeling framework based on the approaches of superpixel-based segmentation and unsupervised feature learning with GAN. We can extract the deep semantic features from our large-scale satellite imagery and generate a set of high-quality synthetic data simultaneously. The computation efficiency in the pixel labeling stage is fully considered, and we are able to obtain very competitive classification results compared to the state-of-the-art methods.

III. Dense Labeling Framework

We aim to label all the pixels on the large-scale VHR images with a small amount of the ground-truth data. The labeling categories contain different terrain objects of the human settlement and natural landscape, such as small houses, large buildings, trees, waterbody, *etc.* The main steps of our dense labeling framework can be summarized into the following steps:

1) *Training GAN with the dense sampling patches:* We densely crop the small subimages with the same patch size from the whole image and feed them as the input to the GAN. The CNN-based discriminator and generator can be trained in an alternative unsupervised way until a certain level of accuracy is reached.
2) *Tessellating the image data into homogeneous superpixels:* We tessellate the image data into irregular but coherent regions, *i.e.* superpixels and then use them as the foot stone of the labeling work. It is very helpful to reduce the data processing complexity and even improve the prediction accuracy near the object boundaries.
3) *Extracting the superpixel features with the GAN discriminator:* We crop the patches from the center of each superpixel and adopt the GAN discriminator as a feature extractor to generate representational features from them. Usually these centralized patches can provide the most useful information for distinguishing different categories.
4) *Training the classifier to label the pixels based on the segmentation hierarchy:* A robust supervised or semi-supervised classifier can learn from the limited ground-truth training data and accurately assign the label to the superpixels. We percolate down the superpixel-wise labels through the segmentation hierarchy, so the pixels inside the same superpixel will finally get the same label. In this way, we can avoid directly labeling the huge amount of pixels and thus improve the computational efficiency of the framework.

We provide the details of the implementation approaches for the first three steps in the following subsections.

A. Unsupervised GAN Architectures

We adopt the CNN-based GAN in our framework since it has been proved more effective for the image data in computer vision tasks [11], [12]. Its architecture mainly involve two models—generator and discriminator, which are denoted as G and D respectively:

- The generator G is used to learn the distribution over the real image data x_r, so it can generate the fake samples $x_g = G(z; \theta_g)$ by mapping the input noise variable z to the data space, where z obeys a prior noise distribution $p_z(z)$ and θ_g represents the multi-layer perceptron parameters of G.
- The discriminator D, as the adversary of G, is developed to distinguish between the samples drawn from the real data x_r and the fake data x_g. It is able to produce the probability score of a sample x as $D(x; \theta_d)$, where θ_d represent the multi-layer perceptron parameters of D, so the score can indicate how similar the sample looks to be drawn from the real data x_r.

These two models are always trained in an alternative way. When we train D, the weights of G are fixed. We need to maximize the score of $D(x)$ for the real image and minimize the score for the fake image simultaneously. When we train G, we need to fix the weights of D, so that the score of $D(G(z))$ can be minimized to fool D. Therefore, the objective function of GAN is written as:

$$\min_G \max_D \mathbb{E}_{x \sim p_r(x)}[\log D(x)] + \mathbb{E}_{z \sim p_z(z)}[\log(1 - D(G(z)))] \tag{1}$$

In an ideal case, we can expect this contesting game between the G and D will be finished with a win-win situation where G is able to generate "realistic" fake samples and D is capable of perceiving the small difference between the fake and real samples. The sensitivity of the D is also helpful for us to extract discriminating features for the terrain objects, since the visual difference occurs at colors, texture, shapes and *etc.* It is very useful for us to accurately distinguish the common terrain categories and even the similar-looking objects.

Like the other existing CNN-based models [13], [14], the GAN in our framework also requires huge amount of image data to train the large number of parameters in its architectures. However, as the Eq. 1 shows, the learning process of GAN is unsupervised, so providing the label information of the ground-truth images is not as necessary as the other models. Since the entire image is always available during the training and its large-scale area contains rich information of the terrain objects, we can therefore crop an adequate number of image patches in a dense sampling way and use them as the training set of our GAN. Fig. 2 shows a toy example of the training pipeline for the GAN model in our framework.

B. Superpixel Tessellation

We adopt the ultrametric contour map (UCM) [15] to produce the tessellation of the image data. This approach has been widely applied to many image applications such as segmentation, classification and object recognition. It is able to provide high-quality closed contours with the state-of-the-art performance. The regions surrounded by these contours form the homogeneous superpixels, as it shown in Fig. 3.

The major steps of UCM superpixel estimation include extracting the local and global contour information:

- The local information is obtained from the multi-scale features, such as brightness, color, texture and *etc.* These local cues are linearly combined into an oriented value to measure boundary strength, *i.e.* $mPb(x, y, \theta) = \sum_s \sum_i w_{i,s} G_{i,s}(x, y, \theta)$, where $\{w_{i,s}\}$ is a set of weights depending on the channels and scales indexed by i and s, and $G_{i,s}(x, y, \theta)$ is a filter to compute the histogram difference between the two semi-circular areas centered at (x, y) with the angle of θ.
- The global information, or the complementary spectral information, is obtained from the sum of gradient values, *i.e.* $sPb(x, y, \theta) = \sum_{k=1} \frac{1}{\sqrt{\lambda_k}} \cdot \nabla_\theta e_k(x, y)$, where λ_k is the eigenvalue for rescaling and $e_k(x, y)$ is the eigenvector of a neighboring weighted graph [15] constructed from $\max_\theta mPb(x, y, \theta)$.

Then we linearly combine these cues obtained from the original image and the eigenvector images into a globalized probability score to indicate the strength of boundary at (x, y) in the direction of θ:

$$gPb(x, y, \theta) = \sum_s \sum_i \alpha_{i,s} \cdot mPb(x, y, \theta) + \beta_{i,s} \cdot sPb(x, y, \theta) \tag{2}$$

The global closed contours can be extracted after an oriented watershed transform (OWT) algorithm [15] is applied. Note that this perceptual grouping strategy is responsible for making good superpixel segmentation. In Fig. 3, we can observe that the areas with significant variation can be captured by smaller superpixels, while the less variation usually results in larger superpixels.

C. Superpixel Feature Extraction

As we described in Section III-A, D is able to produce distinguishing features for different terrain objects when the GAN is fed with an adequate number of diverse images and trained with an optimal number of epochs. The output of the final convolutional layer is flattened into a long vector and this vector can be used as the feature of the input image.

In order to make an appropriate input for GAN, we need to generate a rectangular patch for each irregular superpixel. These patches must be representational since they need to preserve the important side information (such as color, texture, and shape of the objects) as much as possible. Therefore, an intuitive way is to crop the patches from the center of the superpixels, so we can minimize the noising effects from the neighboring superpixels if they are belonged to the different category.

We show these selected patches on the leftmost map of Fig. 4. It can be seen that for the medium or large sized superpixels, the patches can capture the most characteristic areas of the superpixels, while for the small sized superpixels, though the patches also cover additional areas belonged to the neighborhood, however, these patches will not be affected too much on the feature noises because they are mostly obtained from the human buildings which attend to aggregate together for social purposes. In fact, it is somehow helpful for the classifier to learn the common structure from these additional areas of the neighboring superpixels. Note that we also show the rectangular patches for the training points, which are plotted in different colors on the leftmost map of Fig. 4. They are obtained from the provided pixel coordinates instead of the center of the superpixels. It is helpful to maximize the diversity

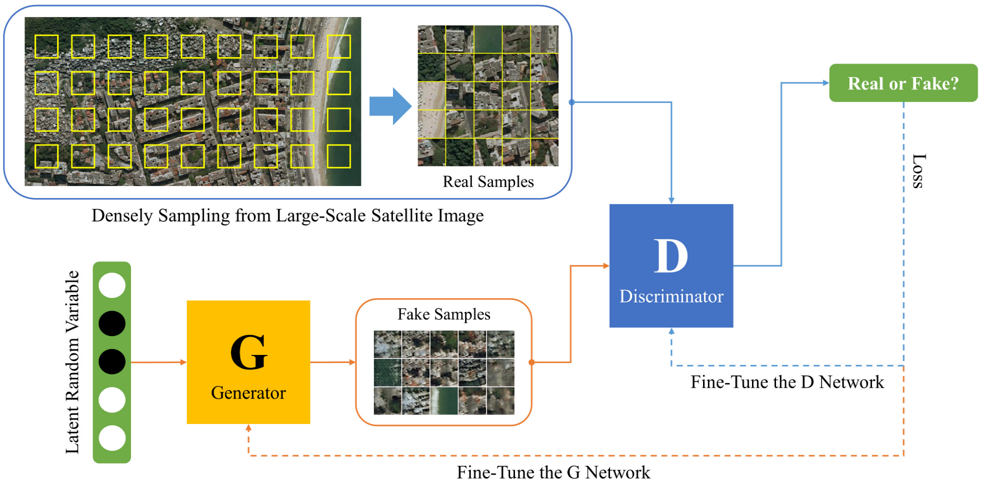


Fig. 2: Illustration of the GAN training pipeline based on a simple dense-sampling strategy.

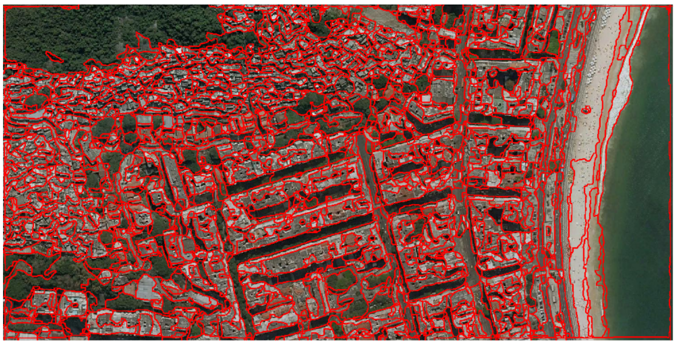


Fig. 3: OTW-UCM superpixel map of Fig. 1.

of the spatially neighboring training samples especially if they belong to the same superpixel on the image.

IV. Experiments

A. Experimental Settings

1) VHR Satellite Dataset: We evaluate our dense labeling framework on the large-scale VHR satellite imagery. Our dataset comprises of images collected from different sources, having varying scales and belonged to three different geographical settings. In Table I, we show the detailed information of these images, including their capture location, image size, and major terrain categories. All of these images contain millions of pixels with a resolution of 1 square meter, so the actual area captured by these images approximately ranges between 1 to 4 square kilometers of the earth surface.

In addition to the major categories that we have listed in the table, there is also a set of diverse regions such as lawn and grass fields, sea sand along the shores, water bodies, farming areas and *etc*. Thus we plus these additional categories to the major terrain categories for each image and we believe this is necessary for an accurate and a comprehensive analysis of our framework.

Fig. 8a shows the ground-truth map of Fig. 1. Different to the limited expert-labeled points that we use for training the superpixel classifier, this map is only used for purpose of accuracy evaluation. In order to eliminate the ambiguity of the pixels located in the mixed regions (*e.g.* sea and sand) or the junction of two areas, we label each pixel with one or more categories—as long as we assign any one kind of the ground-truth categories to the pixel, it is considered as the correct classification. We name this kind of map as *soft ground-truth map* in the following context.

2) GAN Implementation: As Fig. 2 shows, we need to densely sample the rectangular patches from the whole image to train our GAN. In order to guarantee the diversity of these training patches, all the terrain categories should involve no matter a terrain object is fully or partially included. To this end, we crop $10,000$ patches from each image with the same spacing distance and we allow the overlapping of the neighboring patches if the height or width of the actual image is not large enough.

Moreover, we need to set a reasonable value to the cropping patch size, because we may lose the local feature (like the color distribution) of the terrain objects if the patches are too large and the global information (like the object shape and texture) if the patches are too small. Usually, it depends on the *resolution* of the actual satellite imagery. In our case, the patch cropping size is 64×64 and we resize it to 256×256 by the bicubic interpolation method so as to match the input of the GAN.

We follow the architectures in the work of [12] and implemented on TensorLayer [16]. The main advantage is that it can generate diverse fake images (with high resolution) by adding a feature matching loss to the Eq. 1, *i.e.*

$$\left\|\mathbb{E}_{x\sim p_r(x)} f(x) - \mathbb{E}_{z\sim p_z(z)} f(G(z))\right\|_2^2 \quad (3)$$

where $f(x)$ represents the activations on the last three layers of D. It avoids suffering from the mode collapse issues [17] in the training process, and leads to an equilibrium between the learning rate of G and D—otherwise, D may easily get too strong and make G collapse before learning the real data distribution like [11].

After the GAN is trained, we feed the centralized patches (as Section III-C described) to D and extract the flattened superpixel features from the multi-feature layer. A linear SVM classifier is trained with the limited expert-labeled points in the final step of our framework, because it can often yield better classification results than the original CNN architectures [18].

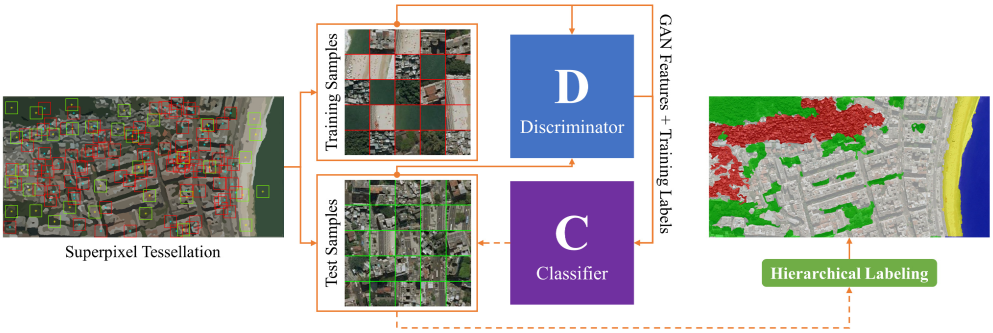


Fig. 4: Illustration of the feature extraction, superpixel classification and pixel labeling steps. A hierarchical labeling method is used at the final step—all the pixels inside the superpixel S will share the label of S that is assigned by the classifier.

TABLE I: Detailed information of the satellite imagery, including the captured location, image size (height and width), major terrain categories and the number of expert-labeled points as the training data.

Image	Captured Location	Size (pixels)	Major Terrain Categories	Expert-Labeled Points (#)
Rio-1,2,3,4,5	Rio, Brazil	$741 \times 1,491$	Slum, Urban, Forest	30, 50, 40, 40, 40
Madison	Wisconsin, USA	$1,807 \times 2,062$	Downtown, Residential	40
Milwaukee		$2,184 \times 2,600$		30
Detroit	Michigan, USA	$1,648 \times 2,305$	Commercial, Residential	40

B. Experimental Results

1) GAN Evaluation: We evaluate our GAN on Fig. 1 and show the performance of its G and D. Examples of the dense-sampling patches are shown in Fig. 5a. The details of the G and D loss are plotted in Fig. 6.

From Fig. 6, we can see both of the curves keep vibrating but they generally move along a stable value—neither G or D loss increases rapidly because the other one becomes too strong. At the beginning, the D loss significantly drops since $G(z; \theta_g)$ have not learned any high-level patterns and D can easily distinguish these noising fake images. After that, G catches up with the learning rate of D and compete with D in an approximately equal strength. Subsequently, D becomes slightly stronger and the D loss is slowly approaching zero.

To further evaluate the generating ability of G, we fix the random z to produce the fake image patches with the G models that are trained with different number of epochs. These patches are shown in Fig. 5b, 5c and 5d. When the number of epochs is small, G only generates some random patterns or learns a few low-level features similar to artificial defects. When the number becomes large, the patches look more “realistic” and more difficult for human to tell the difference from the real patches.

Furthermore, we test the representational ability of the D features with different number of the training epochs. Fig. 7 shows the pixel-based classification accuracies evaluated on Fig. 1. We can observe that the accuracy significantly improves within 20 training epochs. It is consistent with the epoch number that G can generate a batch of reasonable fake samples. Then the accuracy curve fluctuates in a small range around 90%. Generally, it shows the effectiveness of the D features since it can achieves very high accuracy of the pixel labeling, and more evaluation will be conducted on the remaining VHR images in Section IV-B2.

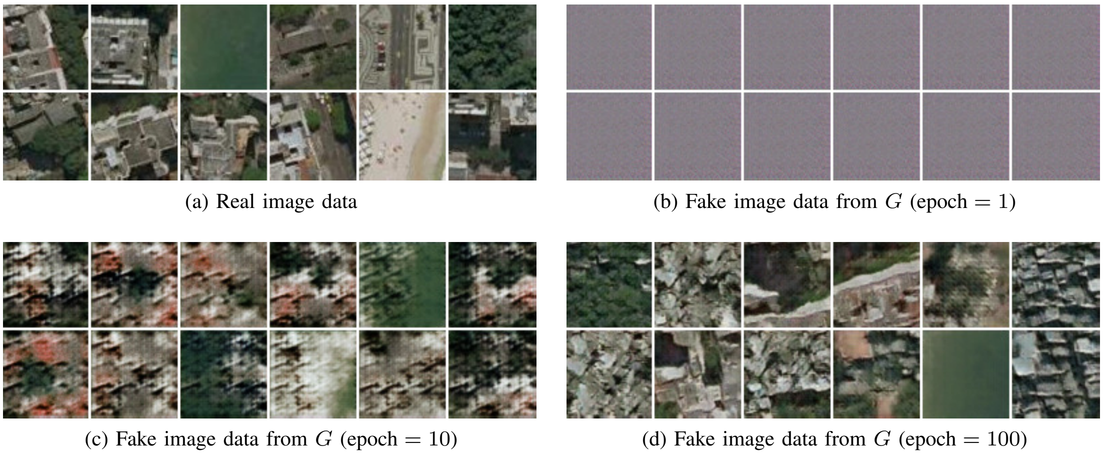
2) Classification Evaluation : We compare our GAN labeling framework against the GMIL algorithm [2] and three other approaches based on heuristic descriptors [4], [19] (denoted as SMLA), sparse auto-encoder features [9] (denoted as SAE) and transfer learning techniques [8] with VGG-16 model [13] (denoted as TL-VGG). These methods have showed good improvement over the standard classification schemes. We show their classification results in Fig. 8.

TABLE II: Pixel-based classification accuracies (%) for the VHR images from Brazil and USA. *Rio-2* is selected as the example in Fig. 1 because it contains the maximum number of the different categories. The last row shows the average accuracy value of each method over all the images.

Images	GMIL	SAE	SMLA	TL-VGG	GAN
Rio-1	77.30	68.62	83.32	**88.52**	86.66
Rio-2	75.83	70.22	88.65	87.66	**92.33**
Rio-3	68.27	63.23	**88.49**	86.11	87.72
Rio-4	72.06	58.44	87.83	**92.32**	87.81
Rio-5	65.35	68.49	85.90	91.20	**91.49**
Madison	69.01	66.81	**85.66**	84.89	83.64
Milwaukee	**84.45**	57.48	76.00	78.08	74.20
Detroit	69.01	65.01	66.85	**83.02**	81.98
AVERAGE	72.66	64.79	82.84	*86.47*	*85.73*

From Fig. 8, we can observe that the results of GAN and TL-VGG look much better than GMIL and SAE. It indicates the great advantage of using CNN architectures in the unsupervised feature learning models. In Table II, we report the classification accuracies on the whole VHR satellite imagery. The classification results of the exemplary VHR images based on our GAN labeling framework are shown in Fig. 9 and 10. Similar to the results in Fig. 8, our GAN and TL-VGG achieve higher *average accuracy values* than the other approaches. Note that the geographical coordinate of *Milwaukee* is a dominant factor to distinguish the small and large buildings and thus this image becomes quite advantageous to the GMIL.

Note that SMLA is the most similar approach to our GAN labeling framework, because the features are directly


(a) Real image data

(b) Fake image data from G (epoch $= 1$)

(c) Fake image data from G (epoch $= 10$)

(d) Fake image data from G (epoch $= 100$)

Fig. 5: Example of the representational images. (a) 12 patches sampled from the real image. (b)(c)(d) 12 fake patches generated from G with different number of training epochs.

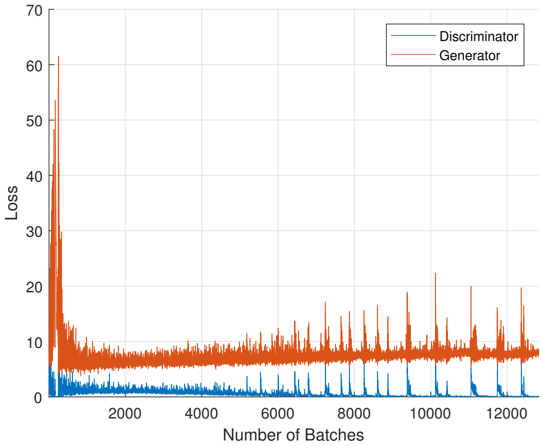


Fig. 6: Details of the loss of D and G that are trained with the image data from Fig. 1. Note that the learning ability of D and G cannot be directly compared by the numerical values. One additional term of Eq. 3 is counted toward the G loss.

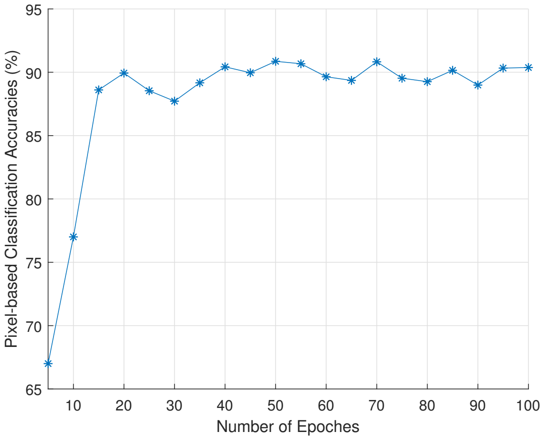


Fig. 7: Classification accuracies (%) of using D in the dense labeling work for Fig. 1. The D models are trained with different number of epochs and then employed to extract the superpixel features for the subsequent SVM.

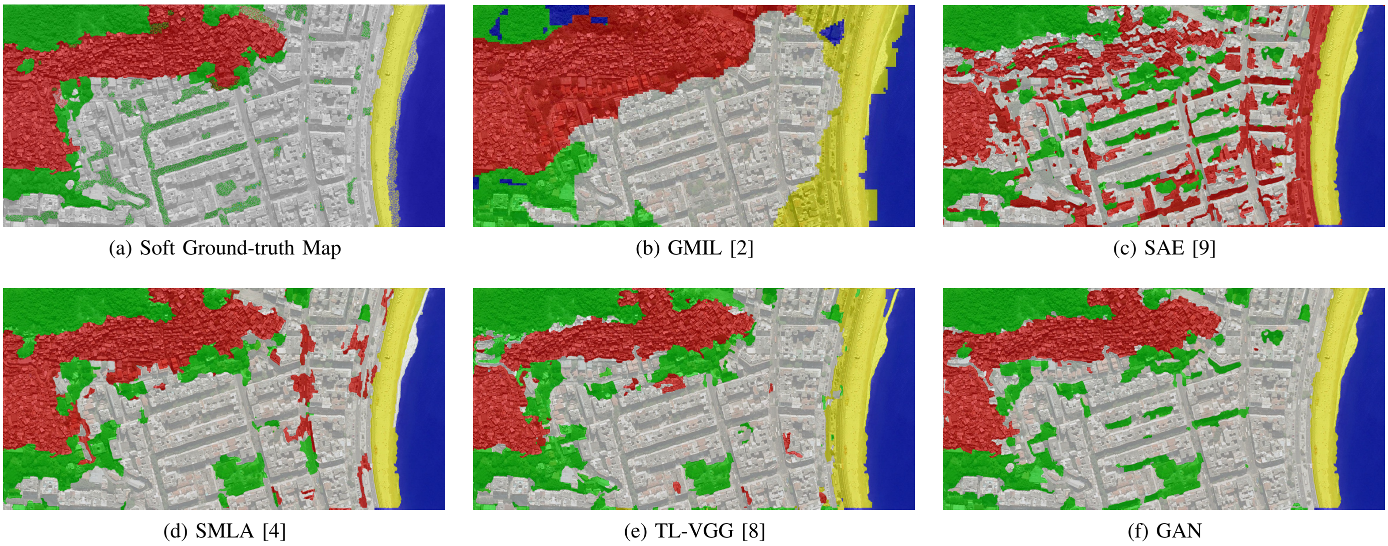
but manually generated by the experienced researchers from the remote sensing field. These features include intensity histogram, corner density, texton descriptors [4]. The performance of SMLA is quite competitive to our GAN method on the VHR images, but GAN works much better on the image like *Detroit*, where the whole image color shifts to green and it thus significantly affects the discriminating ability of the heuristic features. More than 15% accuracy difference can be observed in such situation and it demonstrates the GAN approach is more robust to the color shift.

Furthermore, although TL-VGG can beat our GAN method on 5 of 8 images, however, it is not an “completely” unsupervised approach like GAN, because its powerful CNN model is pre-trained from the large benchmark dataset [20] in a supervised way. The CNN model is always more complicated than D and requires to train on a huge number of *labeled* images with *similar* visual patterns to our terrain objects. This constraint can sometimes make it hard for establishing a transferable CNN model since it never learns the specific information from the current image base. A good example is that GAN significantly outperforms TL-VGG on *Rio-2* even though the its architecture is much simpler than the VGG-16 model [13] in TL-VGG method.

We finally integrate the approaches of our GAN and TL-VGG into one framework (denoted as GAN-VGG), since they can learn the features in very different ways. A reasonable combination can result in a stronger discriminating ability. We train the SVM classifiers of GAN and TL-VGG separately and then take the average of the output probability scores. Each superpixel is labeled with the category that has the the maximum score. It is like we employ two experts and make them vote for the final classification results. Fig. 11 shows the labeling accuracies of these three frameworks. We can see that GAN-VGG always outperforms the other two approaches and it has an average of 3% improvement for each VHR image. It demonstrates that GAN can make important contribution to the labeling work. The high-level features learned by GAN and VGG are complementary to each other.

V. Conclusion

In this paper, we have presented an effective machine learning framework to densely label the pixels in large-scale satellite images. Superpixel segmentation can help us overcome the


(a) Soft Ground-truth Map

(b) GMIL [2]

(c) SAE [9]

(d) SMLA [4]

(e) TL-VGG [8]

(f) GAN

Fig. 8: Classification results of Fig. 1 using different approaches. Different colors are used to present the various terrain categories, *i.e.* red, white, green, yellow, and blue for slum, urban, forest, sand and sea respectively.

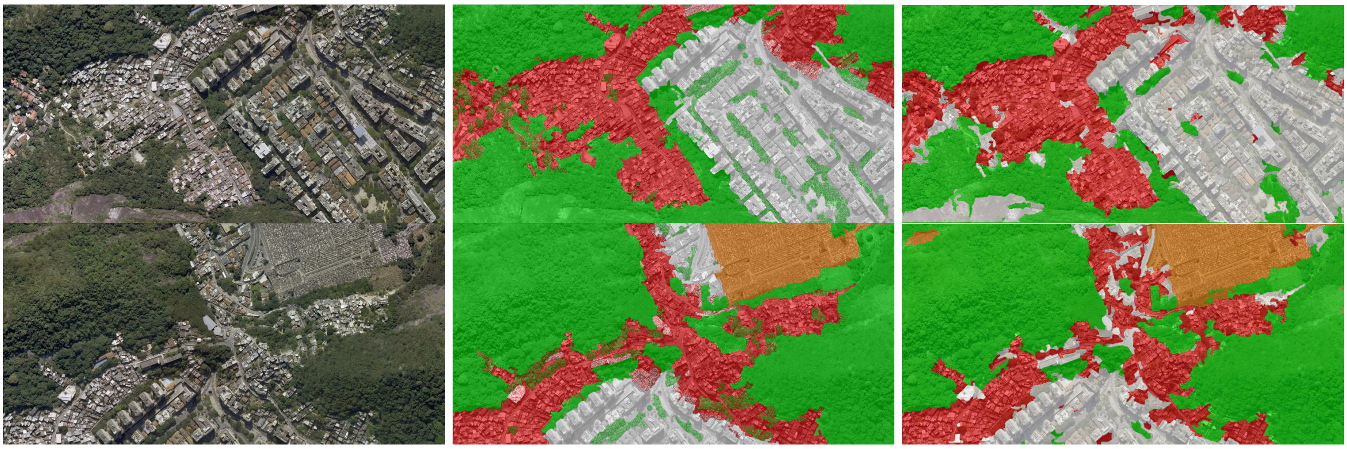


Fig. 9: Example results for the images from Brazil. The left, middle, and right column, respectively, correspond to the actual image, soft ground-truth maps, and SVM classification results. Different colors such as red, white, green, yellow, brown and blue are used to identify the slum regions, urban regions, forests (or large trees), beach sand, farm fields and sea.

big problem of the lack of available ground-truth data with the discriminator in a generative adversarial network (GAN) able to extract the representational features for the subsequent classification. In sharp contrast to previous work, we are able to learn the high-level features in a completely unsupervised way based on the GAN. A simple dense-sampling strategy can provide an adequate amount of unlabeled data for training. We also combine the discriminating abilities of the GAN and the pre-trained CNN model in our labeling framework. We observe a further 3% improvement of pixel-based accuracy in the experiments. In future work, we will focus on improving the realism of fake samples so as to create a comprehensive synthetic dataset. Based on the success of this work, we think it will be helpful for training remote sensing models in other studies, thereby avoiding the need for time consuming and expensive annotations.

References

[1] Y. Yan, X. Tian, L. Yang, Y. Lu, and H. Li, "Semantic-spatial matching for image classification," in *Multimedia and Expo (ICME), 2013 IEEE International Conference on*. IEEE, 2013, pp. 1–6.

[2] R. R. Vatsavai, "Gaussian multiple instance learning approach for mapping the slums of the world using very high resolution imagery," in *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2013, pp. 1419–1426.

[3] M. Sethi, Y. Yan, A. Rangarajan, R. R. Vatsavaiy, and S. Ranka, "An efficient computational framework for labeling large scale spatiotemporal remote sensing datasets," in *Contemporary Computing (IC3), 2014 Seventh International Conference on*. IEEE, 2014, pp. 635–640.

[4] M. Sethi, Y. Yan, A. Rangarajan, R. R. Vatsavai, and S. Ranka, "Scalable machine learning approaches for neighborhood classification using very high resolution remote sensing imagery," in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2015, pp. 2069–2078.

[5] Y. Yan, M. Sethi, A. Rangarajan, R. R. Vatsavai, and S. Ranka, "Graph-based semi-supervised classification on very high resolution remote sensing images," *International Journal of Big Data Intelligence*, vol. 4, no. 2, pp. 108–122, 2017.

[6] W. Hu, Y. Huang, L. Wei, F. Zhang, and H. Li, "Deep convolutional neural networks for hyperspectral image classification," *Journal of Sensors*, vol. 2015, 2015.

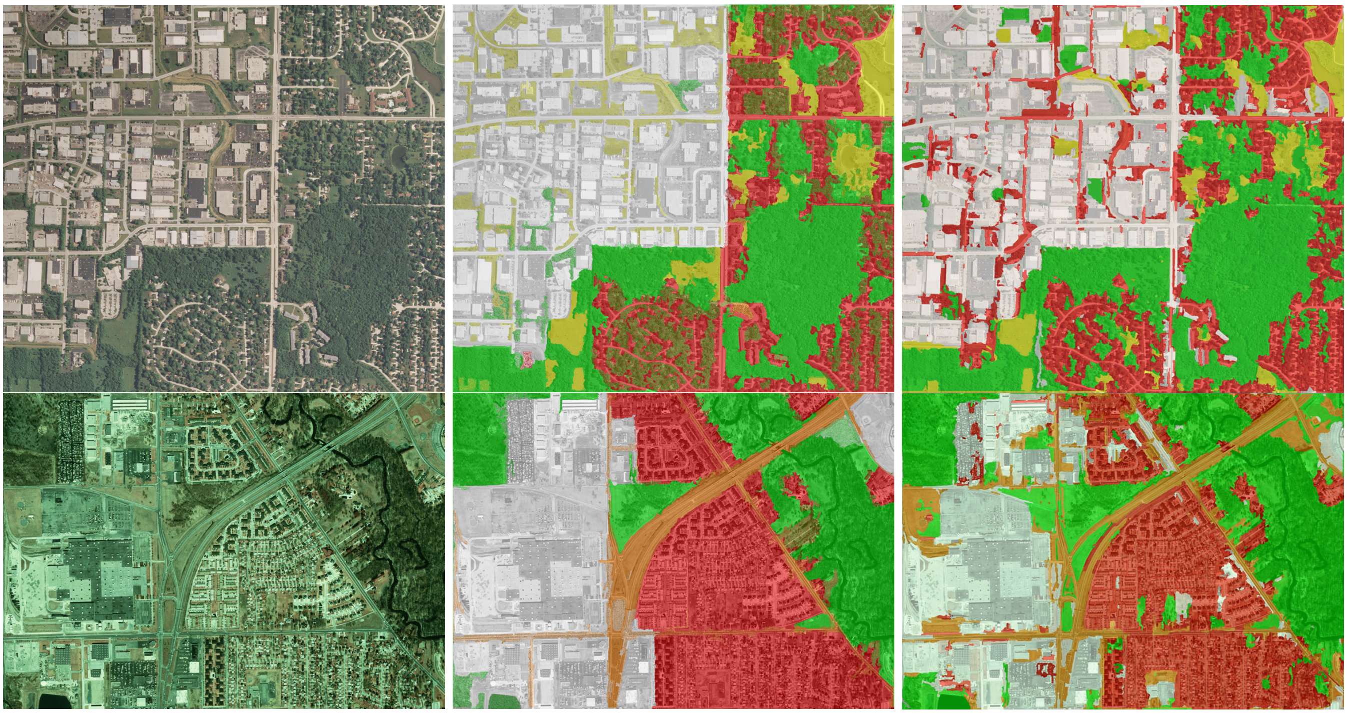


Fig. 10: Example results for the larger images from USA. The left, middle, and right column, respectively, correspond to the actual image, soft ground-truth maps, and SVM classification results. Different colors such as red, white, green, yellow, blue, *etc.* are used to identify the residential areas, commercial areas (or downtown), forest (or large trees), grass (or lawn) and main roads (or highway).

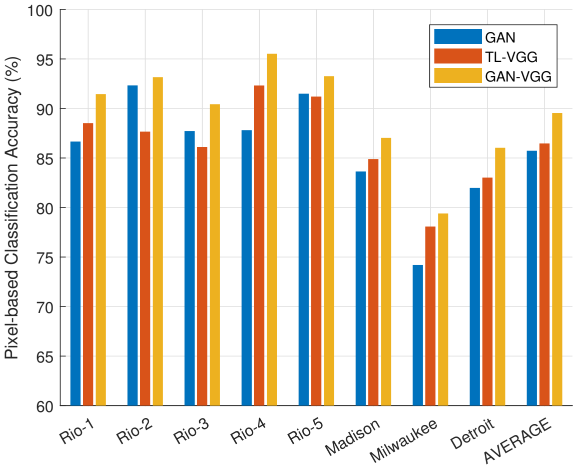


Fig. 11: Comparison of the labeling accuracies (%) of GAN, TL-VGG and GAN-VGG methods on VHR satellite imagery. The integrated GAN-VGG method has 3% further improvement over the original two methods working individually.

[7] M. Castelluccio, G. Poggi, C. Sansone, and L. Verdoliva, "Land use classification in remote sensing images by convolutional neural networks," *arXiv preprint arXiv:1508.00092*, 2015.

[8] O. A. Penatti, K. Nogueira, and J. A. dos Santos, "Do deep features generalize from everyday objects to remote sensing and aerial scenes domains?" in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2015, pp. 44–51.

[9] J. Xu, L. Xiang, R. Hang, and J. Wu, "Stacked sparse autoencoder (ssae) based framework for nuclei patch classification on breast cancer histopathology," in *Biomedical Imaging (ISBI), 2014 IEEE 11th International Symposium on*. IEEE, 2014, pp. 999–1002.

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

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

[12] D. Lin, K. Fu, Y. Wang, G. Xu, and X. Sun, "MARTA GANs: Unsupervised representation learning for remote sensing image classification," *IEEE Geoscience and Remote Sensing Letters*, vol. 14, no. 11, pp. 2092–2096, 2017.

[13] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[14] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," *arXiv preprint arXiv:1602.07261*, 2016.

[15] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 5, pp. 898–916, 2011.

[16] H. Dong, A. Supratak, L. Mai, F. Liu, A. Oehmichen, S. Yu, and Y. Guo, "Tensorlayer: A versatile library for efficient deep learning development," in *Proceedings of the 2017 ACM on Multimedia Conference*. ACM, 2017, pp. 1201–1204.

[17] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training gans," in *Advances in Neural Information Processing Systems*, 2016, pp. 2234–2242.

[18] A. Sharif Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, "CNN features off-the-shelf: an astounding baseline for recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2014, pp. 806–813.

[19] Y. Yan, M. Sethi, A. Rangarajan, and S. Ranka, "Super-scalable computation framework for automated terrain identification," in *Dependable, Autonomic and Secure Computing, 15th Intl Conf on Pervasive Intelligence & Computing, 3rd Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress (DASC/PiCom/DataCom/CyberSciTech), 2017 IEEE 15th Intl*. IEEE, 2017, pp. 1127–1134.

[20] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.