Road Segmentation using UNet and cGAN

Hikaru Hotta, Jerry Qu, Michelle Julia Wai Leng Ng
Stanford University
Department of Computer Science
hhotta@stanford.edu, jerryqu@stanford.edu, mchjl@stanford.edu

Abstract

Road segmentation from satellite imagery is an exciting direction for research with wide ranging impacts. In this paper, we propose a novel neural U-Net architecture to perform road segmentation and use a Pix2Pix cGAN (conditional GAN) to post process outputs. The result of this is that roads can be identified more clearly and accurately than earlier approaches. We train and test this network against a public dataset of roads in Massachusetts.

1. Introduction

The interconnected society we live in is highly dependent on physical infrastructure like roads – the combined length of all the roads on our planet is about 33.5 million kilometres. Obtaining a good ground truth mapping of road locations may require significant location data, or mapping roads manually, which is an enormous task. This is especially problematic in rural areas and less developed areas.

In this paper, we will be investigating automatic road network generation using satellite imagery. Our aim is to develop a deep neural network that is robust enough to not only generate segmentations of roads from both urban and rural satellite images. This could inform the ways in which less developed countries conduct urban planning and infrastructure development; this technology makes wide-spread applications in navigation more accessible. Our input will be a satellite image and our output will be an image of the same size, in which roads are white pixels and everything else are black pixels.

The fundamental challenge of this problem lies in the nature of these images, which may have partially or fully occluded roads or poor image resolution. These challenges may make some roads difficult even for a human to correctly segment. Many prior approaches use a powerful model, which still produces road maps that may be noisy and prone to error and significant post-processing may be needed to produce an accurate road map [11]. Other approaches use an iterative search process to reduce noise [1] while using a less powerful model. In our approach, we propose using a UNet-based model to generate outputs, which we post-process with a Pix2Pix cGAN. We find that this approach both eliminates noise and solves the problem of inferring road connectivity.

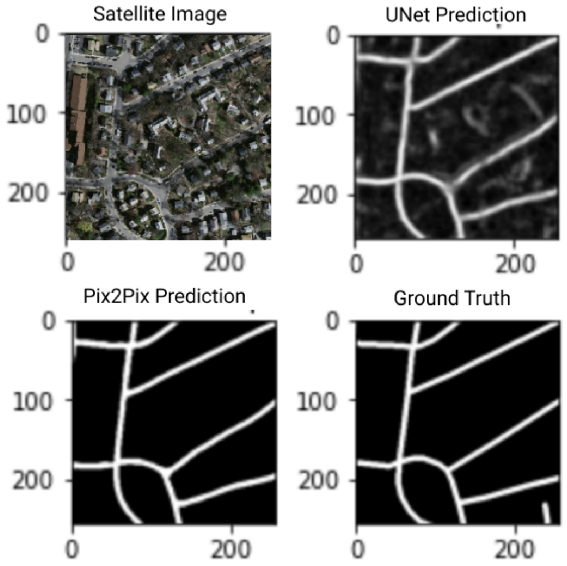

Figure 1. The input satellite image, the segmentation mask predicted from our UNet model, the denoised segmentation mask from our Pix2Pix cGAN, and the hand labelled ground truth segmentation mask.

2. Related Works

One of the first works in road segmentation using a deep neural network was by Minh and Hinton, in which they identified the need to perform post-processing on model predictions to improve performance [6]. Later works used CNNs to perform road extraction, achieving better results [8][13]. In their paper, Zhang et al. used a U-Net architecture as their model, which outperformed prior models [11]. However, CNNs can still produce highly noisy outputs, due

to occlusions from trees and buildings, as well as shadows and other factors. The CNNs generally produced hazy outputs in regions of the image in which the model is unsure, similar to the UNet prediction in Fig. 1. These approaches improved performance by using a more powerful model, but did not solve the problem of noise in generated outputs.

Various other works take a different approach to solve the problem of noisy outputs from neural models. When roads are highly occluded, the model may only detect parts of a road, and simple post-processing may still result in unconnected roads. DeepRoadMapper uses various heuristics to reason about road connectivity, but is limited in effectiveness when road connectivity has moderate to high error [3]. Another approach uses conditional random fields to model road structure and approaches the problem as a graph construction problem [10]. RoadTracer uses an iterative search process guided by a CNN to solve the problem of noisy roads, and performs very well, especially at identifying intersections in roads [1]. These approaches apply specific techniques to solve the problem of denoising, with a focus on inferring road connectivity.

Our approach applies a UNet for the model, similar to the approach in [11]. We then apply a Pix2Pix cGAN to automatically infer road connectivity, instead of manually applying graph heuristics. Our UNet architecture is similar to the one proposed in [7]. The cGAN we use to post-process model outputs is based on earlier works in [2] and [4].

3. Methods

To conduct semantic road segmentation, we trained a UNet architecture [7] on the Massachusetts Roads Dataset. Once we derived segmentation masks from our UNet model, we trained a Pix2Pix cGAN for post-processing using our predicted masks and ground truth masks to improve the performance of our semantic segmentation pipeline.

3.1. Network Architectures

Our UNet architecture consists of a contracting and expanding path. The contracting path is composed of four convolutional units. Each contains two applications of 3×3 convolutions followed by a ReLU which are succeeded by a 2×2 max-pooling operation. The bridge of our UNet contains units with the lowest resolution. It contains two applications of 3×3 convolutions followed by a ReLU which are succeeded by a 2×2 up-convolution operation. The expansive path is composed of four convolutional units. Each concatenates the up-sampled output of the previous unit with the corresponding feature map of the contracting path. 3×3 convolutions followed by a ReLU are applied twice. This is succeeded by a 2×2 up-convolution operation. In our final layer, a 1×1 convolution is applied to reduce the dimensionality to our desired channel.

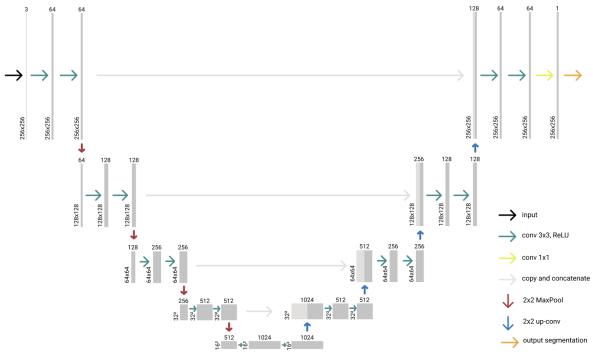

Figure 2. Architecture of our UNet Model (depth = 5). Each block represents a multi-channel feature map with the the number of channels denoted at the top of the block. The height and width are denoted at the bottom left of each block.

Unit	Layer	Filter	Stride	Padding	Output Size
input					256×256× 3
contracting	conv1	3×3/64	1	1	256×256×64
	conv2	3×3/64	1	1	256×256×64
	maxpool1	2×2/64	2	0	128×128×64
	conv3	3×3/128	1	1	128×128×128
	conv4	3×3/128	1	1	128×128×128
	maxpool2	2×2/128	2	0	64×64×128
	conv5	3×3/256	1	1	64×64×256
	conv6	3×3/256	1	1	64×64×256
	maxpool3	2×2/256	2	0	32×32×256
	conv7	3×3/512	1	1	32×32×512
	conv8	3×3/512	1	1	32×32×512
	maxpool4	2×2/512	2	0	16×16×512
bridge	conv9	3×3/1024	1	1	16×16×1024
	conv10	3×3/1024	1	1	16×16×1024
	conv.T1	2×2/1024	2	0	32×32×1024
expanding	conv9	3×3/512	1	1	32×32×512
	conv10	3×3/512	1	1	32×32×512
	conv.T2	2×2/512	2	0	64×64×512
	conv11	3×3/256	1	1	64×64×256
	conv12	3×3/256	1	1	64×64×256
	conv.T3	2×2/256	2	0	128×128×256
	conv13	3×3/128	1	1	128×128×128
	conv14	3×3/128	1	1	128×128×128
	conv.T4	2×2/128	2	0	256×256×128
	conv15	3×3/64	1	1	256×256×64
	conv16	3×3/64	1	1	256×256×64
output	conv17	1×1/1	1	0	256×256×2

Table 1. Network structure of our UNet implementation.

While the segmentation masks predicted by our UNet model capture the structure of most roads, they were generally noisy as seen in Figure 3. While thresholding reduced noise significantly, its effectiveness varied between predicted masks.

Therefore, we proposed a Pix2Pix cGAN to take a deeplearning based approach to denoising. After experimenting with decoder-encoder architectures with and without skip connections, we used an architecture identical to our UNet Model in Figure4 for our generator with the only difference being the input has one channel instead of three. For our discriminator, we decided to use a patchGAN ar-

chitecture (Figure 5). Unlike a traditional CNN architecture which maps images to scalars using a flatten operation, patchGAN outputs an $N \times N$ array which classifies each $N \times N$ patch in the input as real or fake. [2].

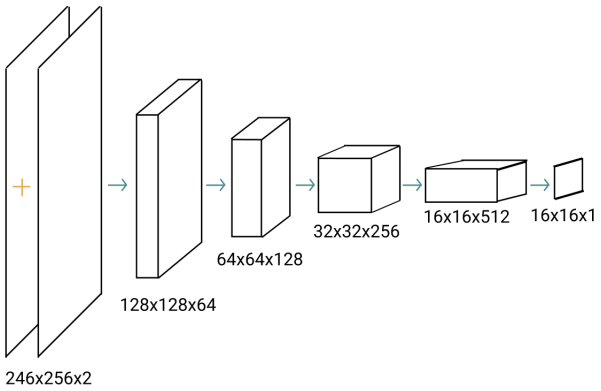

Figure 3. Architecture of PatchGAN used as the discriminator for our Pix2Pix conditional GAN. + represents concatenation while → represents a convolution with a kernel size of 4, stride of 2, and padding of one.

3.2. Training

Our model and preprocessing steps were implemented using PyTorch. The cropped 256×256 satellite images and their corresponding 256×256 segmentation masks were used to train our UNet model (Figure 4 and Table 1). Our UNet model was configured with three input channels (RGB) and one output to represents pixels with roads. We used padded convolutions to ensure that the output images were the same size as the input. We aimed to minimize the cross-entropy loss which was computed as the negative log likelihood of the pixel-wise soft-max activation

$$loss(x, class) = -\log\left(\frac{exp(x[class])}{\sum_j exp(x[j])}\right)$$

where $x[j]$ denotes the output score of class j. The learning rate was set to 0.001 and decayed by a factor of 0.1 every 10 epochs. Using an Adam optimizer, our model converged within 40 epochs.

Using our trained UNet Model, we generated predictions on our training set and used them as inputs to our Pix2Pix cGAN.

Our model's loss function is denoted as

$$\mathcal{L}(G, D) = \mathbb{E}[\log D(x, y)] + \mathbb{E}[\log(1 - D(G(x, z)))]$$

where G is the generator loss function and D is the discriminator loss function which both compute mean squared error. x is the prediction by our UNet model, y is the ground truth segmentation mask, and z is a random noise vector [2].

G tries to minimize the above loss function while D tries to maximize it, which gives us the adversarial objective

$$G^* = arg \min_G \max_D \mathcal{L}(G, D).$$

We wanted G to not only fool D but also to produce an output similar to the ground truth. Therefore, an $\mathcal{L}_1$ regularization term was added to out objective function [2].

$$G^* = arg \min_G \max_D \mathcal{L}(G, D) + \lambda \mathcal{L}_1$$

With an Adam optimizer we trained both our Generator and Discriminator using the training arc depicted in Figure 6.

Both models were trained on an NVIDIA Tesla K80 GPU on a Google Cloud Platform Virtual Machine instance.

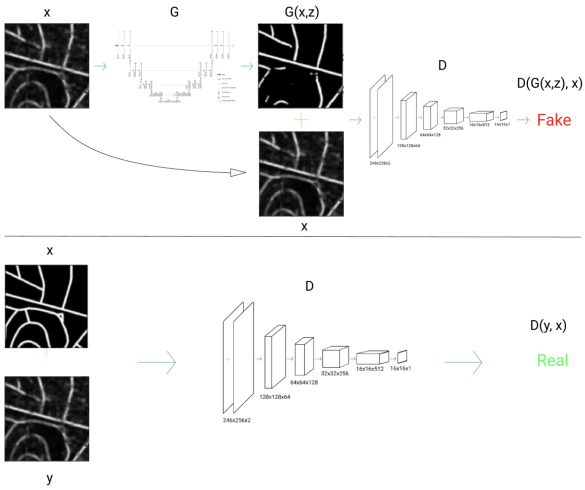

Figure 4. Training a Pix2Pix cGAN. The Discriminator learns to classify $(G(x, z), x)$ and (x, y) tuples as fake and real respectively. As a conditional GAN, both the generator and discriminator observe x, the segmentation mask predicted by our UNet model.

3.3. Evaluation Metrics

To evaluate the performance of our model, we utilized four evaluation metrics on both UNet and UNet + Pix2Pix models.

A. Pixel Error

Pixel error is a naive metric; it is the L1 loss of our prediction compared to the ground truth.

$$PE = \frac{1}{n}\sum_{i=1}^{n} |y - \hat{y}|$$

B. Rand Error

Each object in a segmentation consists of a set of pixels

sharing a common label. The Rand error is the frequency with which the two segmentations disagree over whether a pair of pixels belongs to same or different objects.

$$RE = 1 - \frac{TP + TN}{\binom{n}{2}}$$

C. Jaccard Index

This metric evaluates the percent overlap between the ground truth and the predicted segmentation mask.

$$JI = \frac{\text{prediction} \cap \text{ground truth}}{\text{prediction} \cup \text{ground truth}}$$

D. Precision and Recall

Precision refers to the fraction of predicted road pixels that were ground truth road pixels. Recall refers to the fraction of ground truth road pixels that were predicted to be road pixels.

4. Dataset

The Massachusetts Roads Dataset [5], developed by Mnih et al., contains 151 pairs (139/3/12 training/validation/testing split) of $1500 \times 1500 \times 3$ RGB satellite images and 1500×1500 hand-labelled road segmentation masks. The dataset covers an area of over $2600km^2$ with each pixel corresponding to a $1.2m \times 1.2m$ area. It also encompasses a wide variety of topographies including urban, suburban, and rural regions.

Figure 5. Input (first row) and output (second row) images from the pre-processed dataset.

We pre-processed our data by cropping each image into 256×256 sub-images to reduce the input size of our model. We then transformed our dataset by normalizing each image by the channel-wise mean and standard deviation. After pre-processing, we had 18255, 350, and 1225, images in our training, validation, and test sets, respectively. We did not perform any additional feature extraction on these images, as we believed the images themselves should be sufficient to perform segmentation.

5. Results

5.1. Hyperparameters

For our UNet model, we performed 3-fold cross validation on the learning rate, batch size, and model depth (number of down-sampling blocks). By over fitting the validation set, we selected a learning rate of 0.001, batch size of 16, and depth of 5. For our Pix2Pix cGAN, we performed 3-fold cross validation on the learning rate, batch size, and the probability of the discriminator updating its weights. We selected a learning rate of 0.0002, batch size of 32, and probability of 0.1 which allowed our generator to learn features without being overwhelmed by an aggressive discriminator.

5.2. Quantitative Results

Denoising with Pix2Pix produces noticable improvements in every category, especially in the pixel error, indicating that Pix2Pix cGAN was able to successfully denoise and correct connectivity of roads in most cases. The improvement however was not as large as we expected and we predict that this is because all evaluation metrics besides pixel error were binary classification metrics. To compute these metrics, we thresholded the UNet outputs at 0.5 which could have contributed to a significant amount of denoising.

Metric	**UNet**	**Pix2Pix**	VGG [9]	ResUNet[12]
Pixel Error	0.00385	0.000718	N/A	N/A
Rand Error	0.0271	0.0251	N/A	N/A
Jaccard Index	0.502	0.549	N/A	N/A
Precision	0.839	0.872	0.861	0.919
Recall	0.815	0.822	0.941	0.919

Table 2. Evaluation Metrics on the test set.

While our models had comparable precision to Ventura et al.'s VGG-like model [9] our model fails to match the recall produced by other authors. This indicates that our model could be predicting conservatively relative to comparative models which could be a sign of overfitting.

5.3. Qualitative Results

To evaluate our model qualitatively, we will evaluate on two primary metrics. First, we want to evaluate whether the post-processed outputs show the correct roads visually. Second, we want evaluate whether the road network is correct. That is, whether the output road connectivity matches

that of the ground truth, which is a more difficult condition than the first.

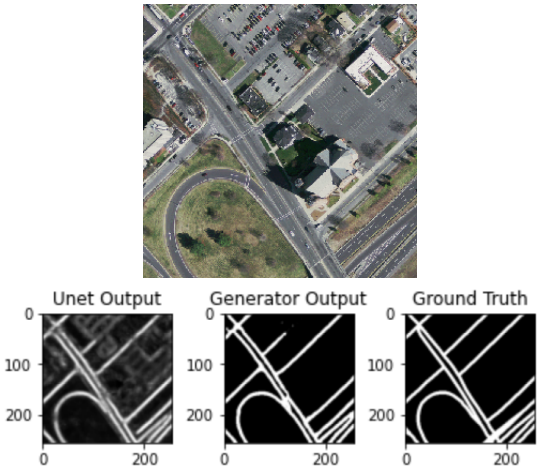

Figure 6. Almost all roads identified correctly, despite presence of a parking lot.

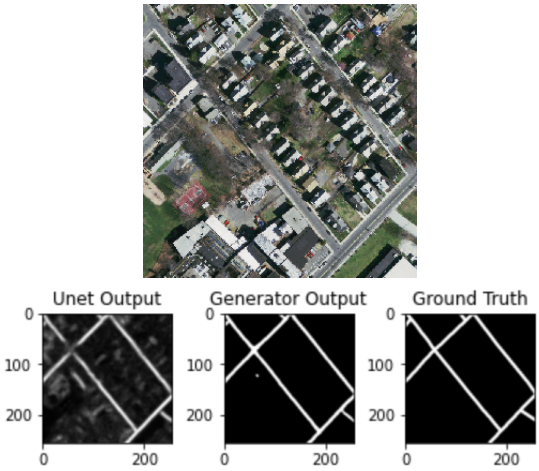

Figure 7. Model was nearly correct and post-processing correctly inferred connectivity.

UNet model outputs generally have correct connectivity with some mistakes. However, the outputs are highly noisy, especially in areas where roads are occluded or unpaved roads. From inspecting the UNet model outputs, we find that in general, the cGAN infers both connectivity and reduces noise correctly. However, in some cases, our model output could allow a human to infer connectivity correctly, but the cGAN incorrectly infers connectivity, an example of which is shown in Fig. 8. There are also some cases where the cGAN is too conservative and removes roads the model is unsure of, rather than keeping them in., as in Fig. 9 However, it consistently performs very well in removing

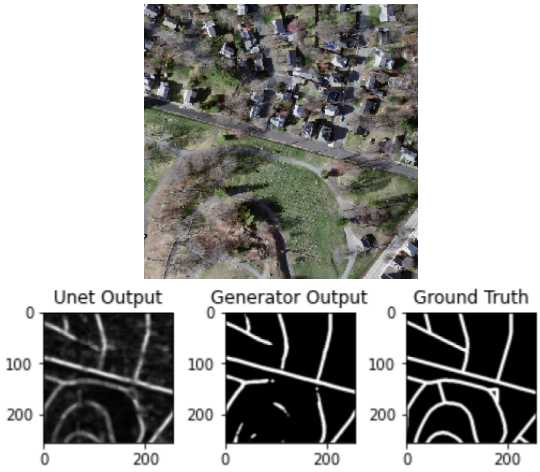

Figure 8. Model was nearly correct but post-processing incorrectly inferred connectivity.

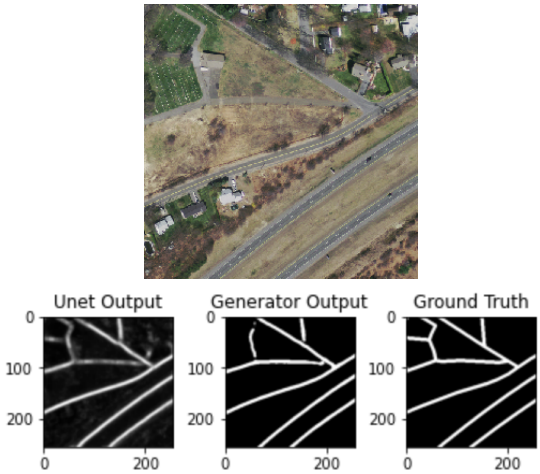

Figure 9. Model was nearly correct but post-processing was too conservative.

the noise from the generated outputs, as shown in Fig. 7.

On the whole, the segmentation masks we produce appear very similar to the ground truth, though for more difficult images, there are some mistakes.

6. Conclusion

In this paper, we present a novel architecture for performing road segmentation. We use a UNet model to generate an intermediate output, then use a Pix2Pix cGAN to perform post-processing on this output to reduce noise and increase road connectivity. The results are promising and mostly infer connectivity correctly. We also find a similar approach without the intermediate output, using just the Pix2Pix cGAN, which uses a UNet architecture in its gen-

erator network, does not perform as well.

We find that there are cases where the cGAN produced results that differ significantly from the ground truth. Future work could involve further refining the cGAN and investigating alternative architectures. We can also better tune our model precision and loss to achieve the break-even point. We also could have used a more advanced architecture in the UNet model, such as the ResUNet proposed in [11].

7. Contributions & Acknowledgements

In the paper, Jerry drafted the abstract, introduction, related work, dataset, and conclusion sections. Hikaru drafted the methods and results sections, including the figures for our model architecture. Michelle also helped to revise and edit the paper.

All three members worked jointly to produce much of the coding and technical work, doing pair programming. The dataset preprocessing work was led largely by Hikaru. Jerry built a large portion of the UNet model and original cGAN. Hikaru also made modifications to both models and implemented the Pix2Pix cGAN. Michelle made contributions to our metrics. Hikaru trained the model on a GCP VM instance. Michelle took the lead in communicating our findings by planning and preparing our video presentation.

There were no non-CS231N contributors. For our codebase, we utilized in part code from the following repositories and articles:

1. https://medium.com/ai-society/gans-from-scratch-1-a-deep-introduction-with-code-in-pytorch-and-tensorflow-cb03cdcdba0f
2. https://towardsdatascience.com/u-net-b229b32b4a71
3. https://github.com/eriklindernoren/PyTorch-GAN/tree/master/implementations/pix2pix
4. https://github.com/jvanvugt/pytorch-unet/

References

[1] F. Bastani, S. He, S. Abbar, M. Alizadeh, H. Balakrishnan, S. Chawla, S. Madden, and D. DeWitt. RoadTracer: Automatic Extraction of Road Networks from Aerial Images. *arXiv e-prints*, page arXiv:1802.03680, Feb. 2018.

[2] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks, 2016.

[3] G. Máttyus, W. Luo, and R. Urtasun. Deeproadmapper: Extracting road topology from aerial images. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 3458–3466, 2017.

[4] M. Mirza and S. Osindero. Conditional generative adversarial nets. *ArXiv*, abs/1411.1784, 2014.

[5] V. Mnih. *Machine Learning for Aerial Image Labeling*. PhD thesis, University of Toronto, 2013.

[6] V. Mnih and G. E. Hinton. Learning to detect roads in high-resolution aerial images. In *European Conference on Computer Vision*, pages 210–223. Springer, 2010.

[7] O. Ronneberger, P. Fischer, and T. Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation. *arXiv e-prints*, page arXiv:1505.04597, May 2015.

[8] S. Saito, T. Yamashita, and Y. Aoki. Multiple object extraction from aerial imagery with convolutional neural networks. *Electronic Imaging*, 2016(10):1–9, 2016.

[9] C. Ventura, J. Pont-Tuset, S. Caelles, K.-K. Maninis, and L. V. Gool. Iterative deep learning for road topology extraction, 2018.

[10] J. Wegner, J. A. Montoya-Zegarra, and K. Schindler. Road networks as collections of minimum cost paths. *ISPRS Journal of Photogrammetry and Remote Sensing*, 108:128–137, 10 2015.

[11] Z. Zhang, Q. Liu, and Y. Wang. Road extraction by deep residual u-net. *IEEE Geoscience and Remote Sensing Letters*, 15:749–753, 2018.

[12] Z. Zhang, Q. Liu, and Y. Wang. Road Extraction by Deep Residual U-Net. *IEEE Geoscience and Remote Sensing Letters*, 15(5):749–753, May 2018.

[13] Z. Zhang, Y. Wang, Q. Liu, L. Li, and P. Wang. A cnn based functional zone classification method for aerial images. In *2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, pages 5449–5452. IEEE, 2016.