Residual Inception Skip Network for Binary Segmentation

Jigar Doshi
CrowdAI
San Francisco, CA
jigar@crowdai.com

Abstract

This paper summarizes our approach to the Deep Globe Road Extraction challenge 2018*. In this challenge we are tasked to find road networks from satellite images. First, we explain our U-Net type baseline model for the challenge. Second, we explain a new architecture that takes in the lessons from some of the popular approaches that we call Residual Inception Skip Net. Finally, we outline our cyclic learning rate based ensembling approach which improved the overall single model performance and the final solution for submission. Our final model increases the IoU by 3 points over the baseline.*

1. Introduction

The recent increase in availability of satellite imagery has opened new areas of geospatial infrastructure analytics. Road extraction via satellite imagery has the potential to map previously un-mapped areas at scale, rapidly modify maps during times of disaster to help coordinate relief, and track global urban development in more detail. Figures 1-3 below show a sample satellite image, its corresponding ground truth mask, and a dense prediction which can be extracted into a geo-referenced road network.

2. Challenge

This challenges [4] required extracting of road networks from satellite imagery. The provided training dataset consists of 6226 RGB images of size 1024×1024 pixels. The imagery is collected via DigitalGlobe satellites and has 0.5 meter pixel resolution. Thus, each image covers $0.26km^2$ and the entire dataset represents $1{,}632.1km^2$. To put this in perspective, London is about $1{,}600km^2$. This challenge is a binary segmentation task in which the input is an RGB image and the output is a binary prediction mask classifying each pixel in the input image.

Road extraction via satellite imagery poses unique challenges. These challenges include, but are not limited to:

1. **Varying off-nadir angles**: Off-nadir angle is the degree to which a point is not directly beneath a satellites sensors. High off-nadir angles increase the rate at which roads are occluded by buildings.

2. **Ambiguous 'Road' definition**: What constitutes a road is unclear from the vantage point of a satellite. There are many road-like features such as trails, paths, farm roads, country roads, dried waterways, and others that can be confused for roads.

3. **Road occlusion due to trees**: Models must learn to interpolate roads beneath trees when the road itself is not visible.

2.1. Metric

Intersection over Union (IoU) is one of the standard measure performance measure for segmentation. Given a set of images, the IoU measure the similarity between the prediction and the ground-truth per pixel.

$$IoU = \frac{TP}{TP + FN + FP} \tag{1}$$

where TP, FP, FN denote the true positive, false positive and false negative numbers. This score is always between 0 and 1. We call the average scores across the set of images as mean IoU score ($mIoU$). All the results in this paper are $mIoU$ scores. For a more detailed explanation and justification of the metric please refer [4]

3. Models

Empirically, for the task of binary segmentation, it has been shown that using a simple encoder decoder type architecture with skip connection performs quite well [1]. We explore the idea of making modern classification architectures work well for binary segmentation tasks. We use ResNet [5] and Inception style [16, 17] encoder architectures and propose a hybrid architecture called Residual Inception Skip Net. This method improves the baseline models by **3** percentage points and outperforms other models.

Figure 1: Satellite Image

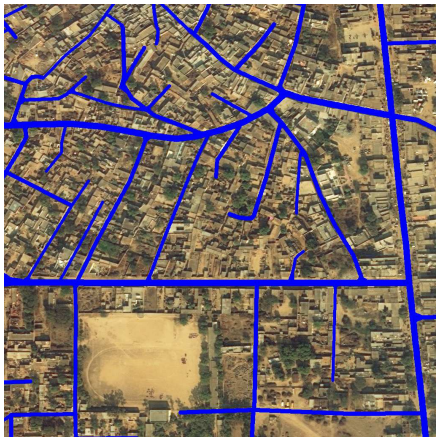

Figure 2: Ground Truth

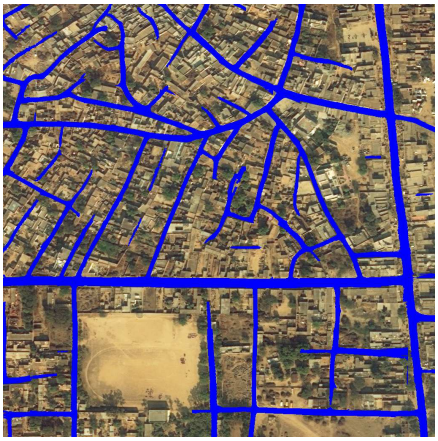

Figure 3: Predicted Mask

3.1. U-Net

Our first model was inspired by the family of U-Net architectures [12], where low-level feature maps are combined with higher-level ones, which enables precise localization. This type of network architecture was designed to effectively solve image segmentation problems, particularly in the medical imaging field. U-Net is generally a default choice for segmentation challenges in Kaggle.

The encoder of the model consists of a VGG network [14] with the addition of batch norm [17] and a total of 5 downsampling layers. The choice of the depth of the network was informed by careful analysis of the dataset, task and the receptive field [10]. We decided to keep a constant number of 128 feature maps throughout the network. This was based on the key observation that we can afford the network to lose some representational power in the encoder half as the model has access to low-level features in the decoder half via the skip connection.

The decoder is similar to the encoder where instead of max-pooling we use deconvolution layers to upsample with a skip connection from the encoder, combining deep representations of the prior decoder layer with more precise spatial representations from the corresponding encoder layer. The final head consists of a sigmoid activation function.

3.2. DeepLab

Our encoder consists of a ResNet block with 55 convolutional layers with 3×3 kernels. We use a convolutional layer with stride 2 after 6 residual blocks forming 13 convolutional layers. This divides the whole encoding network in 4 parts. We use 16, 32, 64 and 128 kernels in each of these parts respectively. This gives us a feature map of 128 dimensions with $1/8$ of the original resolution. The decoder consists of 3 fully convolutional layers with number of kernels 64, 32 and 16 respectively. Each of these layers upsamples its input to be double its resolution. The last convolutional layer converts the feature map into scores followed by a sigmoid activation function. Thus the whole network consists of 55 convolutional layers for the encoder, 3 fully convolutional layers for the decoder, followed by a convolutional layer to output the class labels.[3, 2, 18]

3.3. Residual Inception Skip Net

This model is a convolutional encoder-decoder architecture with the inception modules instead of standard convolution blocks. The inception models are the originally proposed in [8] but with asymmetric convolutions. For example, a 3×3 convolution is replaced with a 3×1 convolution, then batch norm followed by 1×3 convolution. This is better since it reduces the number of parameters and gives similar performance. All weights are initialized with the He norm [6] and all convs are followed by batch norm and then activation. We used a leaky RELU with a slope of $-0.1x$ as our activation function.

Downsampling is performed using a strided convolution followed by batch norm and leaky RELU. Upsampling is done using 2×2 upsampling convolutions. Both upsampling and downsampling layers are followed by a dropout of 0.9. Just like U-Net, we also add a skip connection linking identically sized layers between encoder and the decoder. The connections outputted the sum of the input and a residual block where a 1×1 convolution is followed by batch norm. We experimented with trying to scale down the encoder layer but that resulted in slightly worse performance.

The final head consists of a 1×1 convolution followed by a hard sigmoid activation function. We use this activation function instead of a standard sigmoid because of our choice of a sensitive dice loss function. In the case of soft sigmoid function, the model activation only tends to 0 or 1 values which leaves a lot of small residual error that adds up and diffuses the model's ability to focus on harder error signals.

4. Training Setup

4.1. Data

We normalize all the data in training by subtracting the mean and scaling by the standard deviation.

We apply a host of data augmentations to the training data before feeding them into the model. We augment the images fed into all the models by randomly scaling between $1/1.1$ and 1, jittering the color space, zooming between $1/1.1$ and 1.1, shearing between -5 and 5 degrees. Each image had a 10% chance of having the above augmentations applied. Not surprisingly, adding various data augmentation improved the model performance. However, stronger data augmentation reduced model generalization. We think this might be due to the over-perturbation of training distribution into modes absent in the validation/test distribution.

All the models' input dimension is 256×256. We split each of the 1024×1024 images in the training set into 16 training patches, adding black-fill as necessary. For internal training, we split the dataset into $85 - 15$ for training and validation set.

4.2. Loss Function

The dice similarity coefficient (DSC) measures the amount of agreement between the model prediction and the ground truth. It is a widely used metric in high class imbalance segmentation tasks [11, 13]

The dice similarity coefficient is defined below

$$DSC = \frac{2|P \cap Q|}{|P| + |Q|} \tag{2}$$

where P is the model's segmentation output and Q is the human-annotated ground truth. This function is however not differentiable and therefore cannot be used directly as a loss function for our neural networks.

We use a continuous version of the Dice score that allows differentiation and can be used as a loss function in optimization of our network using stochastic gradient based methods.

$$\mathcal{L}_{DSC} = -\frac{2\sum_i^N s_i r_i}{\sum_i^N s_i + \sum_i^N r_i + \epsilon} \tag{3}$$

where s_i represents a continuous value from the model for each pixel which is typically an output from a $sigmoid$ or $softmax$ activation function. r_i represents the ground truth annotation for each pixel. ϵ is a smoothing factor typically set to 1.0

We experimented with a baseline binary cross-entropy loss and found it to be consistently worse than dice loss across all the configurations.

4.3. Optimization

We trained all the models with ADAM [9] using $\beta 1$ as 0.9 and $\beta 2$ as 0.999 without any weight decay. Unless stated otherwise, we started the training with a learning rate of $2e^{-4}$. We reduce the learning rate by a factor of 2 after observing no improvement in the validation loss for 10 epochs. All the models were trained for 150 epochs.

We fine tuned the model after 100 epochs using the cyclic learning rate (CLR) technique proposed by [7, 15]. Intuitively, the idea is to encourage the model to reach different local minima and then ensemble those checkpoints. The way we achieve this is by cycling the learning rate between a reasonable range of values. When we reach a high learning rate, we hope that the model jumps out of the current local minimum and drifts towards a different and often equivalent local minimum. Subsequently, as the learning rate lowers, we hope the model spirals down a new local minimum. This ensembling technique boosts performance for all models.

5. Ensembling and Prediction

We explored several different ensembling techniques as is common in these types of competitions.

1. **Model prediction averaging:** We average all the model's predictions

2. **Exponential Average of weights:** For each model we use an exponential moving average of the parameters from last few steps to evaluate the model [8].

3. **Cyclic Snapshot Averaging:** We use the above mentioned cyclic learning rate approach to average parameters from all the checkpoints with the lowest learning rate, that is, from the presumed different local minimum.

During the prediction time, for each of the 1024×1024 images, we predict size 256×256 patches at a time with a stride of 64. We normalize the images the same way we normalize the training data. We saw no improvements in model performance when we predicted with data augmentations.

6. Results

In Table1 we share results based on the random 85% and 15% split of the development (training) set. Since the scoring on the test set had limited number of tries we can only share the final submission number, which is also the best model in the development set. Our final score on the hidden test set was 0.61 which was obtained by ensembling of the top 61 best performing model on the development set.

According to Table1 we see a consistent improvement in the performance when using an exponential moving average

Models	Single Model	Exp Avg	CLR
U-Net	58.1	59.0	59.2
DeepLab	58.3	59.3	59.4
ResInceptionSkip	60.1	61.2	61.3

Table 1: Results are in $mIoU$ units. Single model represents no ensembling. Exp Avg represents exponential moving average of parameters over last few steps. CLR represents cyclic learning rate as explained in 5

of checkpoints from several steps as explained in Section 5. When we use cyclic learning rate in conjunction with exponential moving average, we see a consistent boost in performance. Also, we observe that using our proposed hybrid model consistently outperforms the other two baseline models.

7. Future Work

Due to time constrain we were unable to experiment with various post-processing strategies like CRF. These should almost certainly improve the model performance. In addition, it would be interesting to try using a road network specific objective function that would try to optimize directly for connectivity and in turn be more robust around occlusion.

References

[1] Carvana image masking challenge–1st place winner's interview. http://blog.kaggle.com/2017/12/22/carvana-image-masking-first-place-interview/, Dec. 2017. Accessed: 2018-5-1. 1

[2] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Semantic image segmentation with deep convolutional nets and fully connected CRFs. Dec. 2014. 2

[3] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. June 2016. 2

[4] I. Demir, K. Koperski, D. Lindenbaum, G. Pang, J. Huang, S. Basu, F. Hughes, D. Tuia, and R. Raskar. DeepGlobe 2018: A challenge to parse the earth through satellite images. May 2018. 1

[5] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. Dec. 2015. 1

[6] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing Human-Level performance on ImageNet classification. Feb. 2015. 2

[7] G. Huang, Y. Li, G. Pleiss, Z. Liu, J. E. Hopcroft, and K. Q. Weinberger. Snapshot ensembles: Train 1, get M for free. Apr. 2017. 3

[8] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of The 32nd International Conference on Machine Learning*, pages 448–456, 2015. 2, 3

[9] D. Kingma and J. Ba. Adam: A method for stochastic optimization. Dec. 2014. 3

[10] W. Luo, Y. Li, R. Urtasun, and R. Zemel. Understanding the effective receptive field in deep convolutional neural networks. Jan. 2017. 2

[11] F. Milletari, N. Navab, and S.-A. Ahmadi. V-Net: Fully convolutional neural networks for volumetric medical image segmentation. June 2016. 3

[12] O. Ronneberger, P. Fischer, and T. Brox. U-Net: Convolutional networks for biomedical image segmentation. May 2015. 2

[13] C. Shen, H. R. Roth, H. Oda, M. Oda, Y. Hayashi, K. Misawa, and K. Mori. On the influence of dice loss function in multi-class organ segmentation of abdominal CT using 3D fully convolutional networks. Jan. 2018. 3

[14] K. Simonyan and A. Zisserman. Very deep convolutional networks for Large-Scale image recognition. Sept. 2014. 2

[15] L. N. Smith. Cyclical learning rates for training neural networks. June 2015. 3

[16] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. Sept. 2014. 1

[17] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. Dec. 2015. 1, 2

[18] P. Wang, P. Chen, Y. Yuan, D. Liu, Z. Huang, X. Hou, and G. Cottrell. Understanding convolution for semantic segmentation. 2