CS5304 Part 3: DeepGlobe Road Extraction from Satellite Imagery

Jennifer Ding
jd953@cornell.edu

Renran Zhou
rz348@cornell.edu

Abstract

Growing satellite imagery datasets offer a wealth of information on our changing cities and natural environments. Using computer vision, we can automatically identify and localize landmarks such as roads, buildings, and bodies of water to provide an up-to-date data source for a range of use cases such as autonomous vehicles, disaster relief, and climate change analysis. The DeepGlobe challenge provides over 7,000 satellite images for the task of road extraction. Our method explores different data loading techniques and deep learning architectures to improve road detection results.

1 Introduction

As satellite imagery datasets grow, new opportunities arise to analyze the infrastructure of our urban and and natural environments. Computer vision offers a valuable solution to automatically identify landmarks such as roads, buildings, and land/water cover. These continuous studies will be valuable for up-to-date mapping and understanding human and animal influence on different environments as well as the impact of climate change. Other applications include providing an up-to-date city road map for autonomous vehicles or monitoring disaster zones, where obtaining on-the-ground footage is difficult or impossible.

DeepGlobe has identified the Road Extraction challenge for a CVPR-affiliated competition. The task is a binary classification problem where each pixel is identified as a road or not a road. These will be used to generate a mask, which will be compared against labeled road location to evaluate the accuracy of each algorithm.

2 Related Work

Prior work with satellite imagery for road mapping use deep learning methods to perform initial segmentation (Mattyus et al., 2017) as well as labeling and addressing of roads within a satellite image frame (Demir et al., 2018). Based on a previous EPFL Kaggle competition, we will reference methods of using neural networks to create a robust model for road detection (Pavllo et al. 2017). These works also utilize graph-and proximity-based algorithms to improve road labels, allowing for additional detail to be annotated and to identify potential missed segments.

Other improvements on the data and modeling side include exploring deeper CNN architectures and image pre-processing techniques to enhance road-like features before detection.

3 Dataset

The dataset includes 6226 satellite images and their associated road location masks for training and 1243 satellite images for validation. The images (Figure 1) from DigitalGlobes satellite are RGB, size 1024 x 1024, and 50 cm pixel resolution. The masks (Figure 2) are 3 channel RGB, though visually, they show white representing the road location and back representing the background. The mask pixel intensity values may not be 0 or 255 and when converting to labels we must binarize them at threshold 128. Finally, it should be noted that not every road is labeled from the satellite images, as the roads were manually annotated. To reduce complexity of this task, only main roads were annotated. This discrepancy can be seen between Figure 1 and 2.

Figure 1: Satellite Image

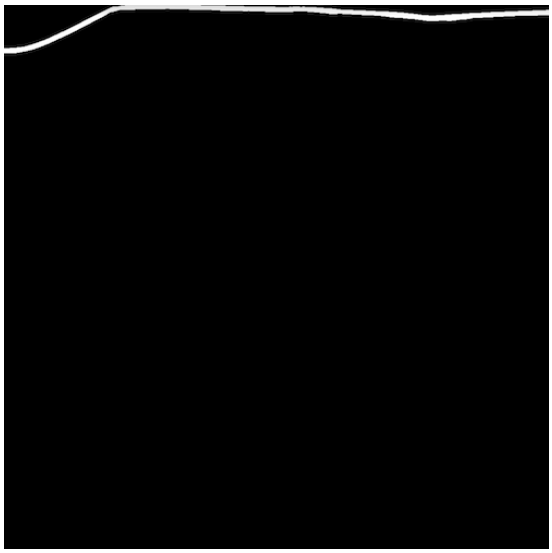

Figure 2: Mask of Road Annotation

4 Baseline Model Description

The baseline model has the end-to-end functionalities of reading in satellite image data, extracting features, training a model with the features and labels from the mask, and finally generating a road segment mask on a test image and comparing that to the ground truth mask.

For each pixel in an image, a window will be generated around the pixel and features extracted from that window. The associated label (road - white or not road - black) will be extracted for that pixel from the mask image. This is the core data point that will be fed into the training model.

For our baseline model, we will use simple image features - RGB mean and standard deviation - and two supervised learning models - Logistic Regression and K Nearest Neighbors - to get a basic understanding of quality of road identification using simplistic methods.

5 CNN Model Description

For our CNN model, we use a Keras model with the following architecture (Figure 3) inspired by Pavllo, et al. to classify images from our dataset. This model takes in a window of set size (ex. 32 x 32 or 16 x 16), and applies three layers of Convolution + Leaky ReLU, Max Pooling, and Dropout before a Fully connected + Leaky ReLU, dropout, and then softmax activation for the binary classification. The model uses an Adam optimizer and binary cross entropy for the loss function.

Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 16, 16, 64)	4864
leaky_re_lu_1 (LeakyReLU)	(None, 16, 16, 64)	0
max_pooling2d_1 (MaxPooling2	(None, 8, 8, 64)	0
dropout_1 (Dropout)	(None, 8, 8, 64)	0
conv2d_2 (Conv2D)	(None, 8, 8, 128)	73856
leaky_re_lu_2 (LeakyReLU)	(None, 8, 8, 128)	0
max_pooling2d_2 (MaxPooling2	(None, 4, 4, 128)	0
dropout_2 (Dropout)	(None, 4, 4, 128)	0
conv2d_3 (Conv2D)	(None, 4, 4, 256)	295168
leaky_re_lu_3 (LeakyReLU)	(None, 4, 4, 256)	0
max_pooling2d_3 (MaxPooling2	(None, 2, 2, 256)	0
dropout_3 (Dropout)	(None, 2, 2, 256)	0
flatten_1 (Flatten)	(None, 1024)	0
dense_1 (Dense)	(None, 128)	131200
leaky_re_lu_4 (LeakyReLU)	(None, 128)	0
dropout_4 (Dropout)	(None, 128)	0
dense_2 (Dense)	(None, 2)	258
activation_1 (Activation)	(None, 2)	0

Total params: 505,346
Trainable params: 505,346
Non-trainable params: 0

Figure 3: CNN Model Architecture

6 Experimental Setup

For the baseline model experiments, 10 images were used to gather a quick understanding of what was possible using supervised learning. For the CNN model experiments, 200 images were used for training and 50 images for testing. Due to the fact that extraction of 256 to 4096 pixel windows per image pixel is slow, memory and computation-intensive processes, the full training dataset was not used, unfortunately.

7 Results

The Baseline Model test showed that the generated road segmentation matched the actual mask annotation with an accuracy of 0.95567 for the K Nearest Neighbor model and 0.972359 for the Logistic Regression Model. However, it became clear that the model was mostly predicting class 0 for every test sample, which is what resulted in the high accuracy for this small test set. This prompted the recognition that the dataset was very unbalanced between classes: 0.96 was class 0 (not road) and 0.04 were class 1 (road). Thus, the CNN model, dataset balancing was performed. Before feeding data into the model, the number of class 1 and class 0 pixels were counted and class 0 pixels were randomly chosen to be discarded so that there was exactly equal data representation between classes.

For the CNN test, greatly improved results were observed as number of images and number of training epochs increased. Experiments were run with a few different window and stride sizes, which led to different number of features we input into the model, largely due to the step of equalizing class count. Results can be seen in Table 1. In terms of training, the loss decreased rapidly in the first few epochs. For the 16 x 16 model, loss jumped from 3.7196 to 0.6132 in 10 epochs. However, the difference between that loss to 0.4213, the final training loss of epoch 200, was quite close, as visualized in Figure 4.

	Number of Windows (Training Data)	Number of Epochs	Training Accuracy
64 x 64 window Stride 64	1690	200	0.7695
32 x 32 window Stride 32	16246	200	0.8071
16 x 16 window Stride 16	74252	200	0.8044

Table 1: Training Accuracy for Different Models

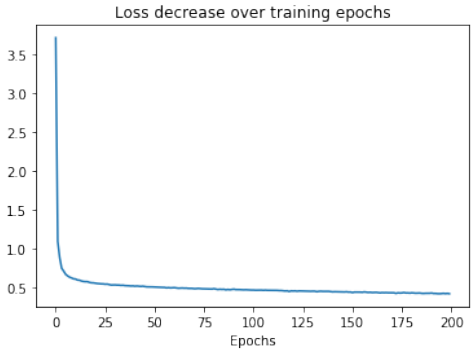

Figure 4: Training Loss for 16 x 16 Model

Finally, the 16 x 16 model is applied to testing data, predicting road pixel location in 50 images, or 198450 windows. As shown in Table 2, the overall accuracy is 0.8105, with most of the errors as false positives, a pixel predicted as road that actually was not road. The false positives occurred 24 times as often as the false negatives. Based on the variation of background in the images and high number of roads present that aren't annotated, this ratio makes sense. These results show that by addressing the false positive problem, major gains in accuracy can be made.

Based on the model's predictions, a predicted mask is generated which shows the road location, intensity inverted in this visualization. The comparison between the real and predicted masks can be seen in Figure 5 and 6. The false positives revealed by the accuracy score can be seen as isolated patches of noise throughout the image and a rectangular patch connecting to the main road. Overall, the position and shape of the predicted mask is comparable to the real mask, though shifted and amidst noisy false positives. Another way to reduce the noise of predictions is to reduce the stride in window generation for training data. However, this also scales of memory and computation load.

	Validation Accuracy	False Positive	False Negative
16 x 16 model	0.8105	36087	1507

Table 2: Test Results for 50 Images

8 Conclusion

Our work explores techniques for extracting road locations from satellite imagery. There is still work to be done in addressing the false positive and negative error types, such as differentiating between different road and background types, which vary by geography. Further work can use blob detection to remove non-contiguous regions that don't have a road-like shape. This second pass could supplement the CNN model predictions. Lastly, in order to effectively utilize the satellite data, it will be essential to develop methods and invest in computing and memory power that can handle this high resolution data. Even still, the results show that deep learning provides a promising opportunity to collect up-to-date mapping data without manual annotation.

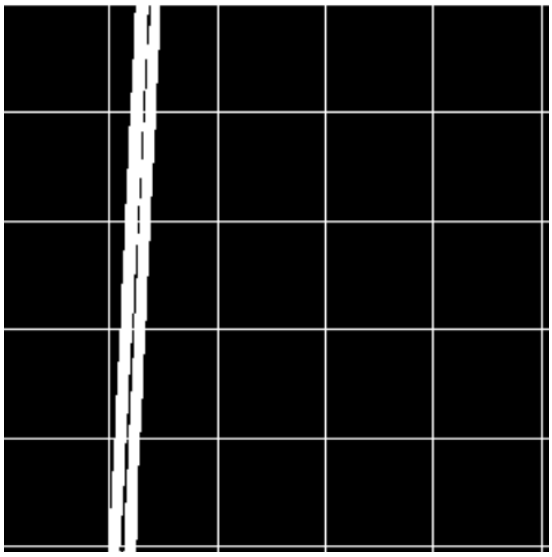

Figure 5: Real Mask

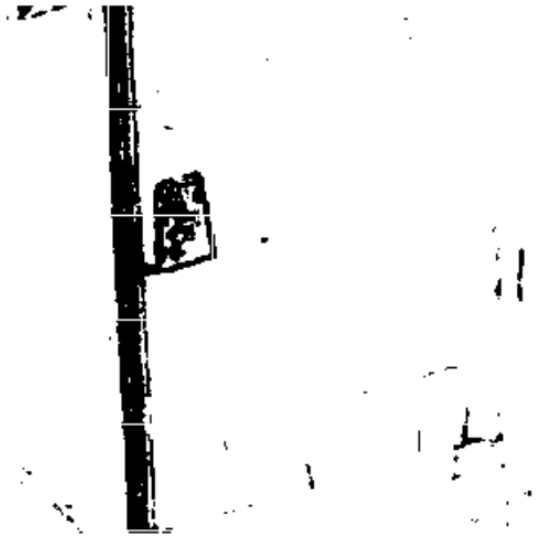

Figure 6: Predicted Mask

Our code can be viewed on Github: https://github.com/dingaaling/roadExtract.

9 References

Demir, I., Hughes, F., Raj, A., Dhruv, K., Muddala, S.M., Garg, S., Doo, B., Raskar, R. Generative Street Addresses from Satellite Imagery. ISPRS Int. J. Geo-Inf. 2018, 7, 84.

Mttyus, G., Luo, W. and Urtasun, R., "DeepRoadMapper: Extracting Road Topology from Aerial Images," 2017 IEEE International Conference on Computer Vision (ICCV), Venice, 2017, pp. 3458-3466.

Pavllo, D., Martinelli, M., and Hwang, C. Satellite Image Segmentation with Convolutional Neural Networks.