Semantic Segmentation of Roads from Satellite Images using Deep Convolutional Neural Networks

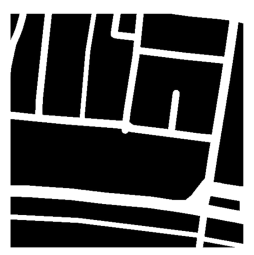
COMP 540 Spring 2019 Term Project
Alex Yang and Corrin Fosmire

Exploratory Data Analysis

- Input Data is thousands of satellite images taken of country, city, and other landscapes.
- Goal: Predict which pixels are roads, and which pixels are not.

Sample Image

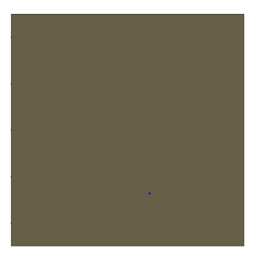
Paired Mask



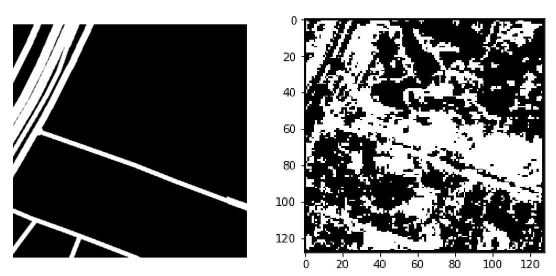
Sample Image for ANN

Average Road Pixel

Average Non-Road Pixel

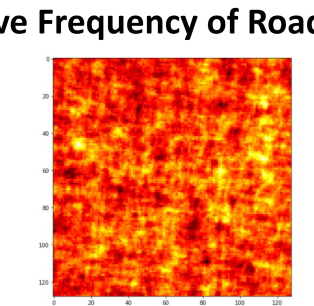


Ground Truth vs Prediction

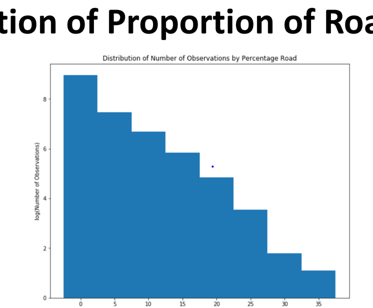


- We observe that the average road pixel is slightly lighter than the non-road pixel, but there isn't an easily identifiable difference
- We attempted a simple ANN model which predicted on 3x3 patches

Relative Frequency of Road Pixels



Distribution of Proportion of Roads Pixels



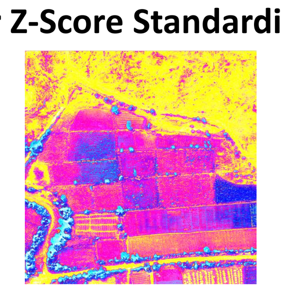
- There are more road pixels on the right side of the images
- There is an exponential dropoff in frequency as the percentage of road pixels increases. Many more examples with very few road pixels!

Preprocessing Steps

- **Mean and Variance Scaling:**
  - Standardizing training data using a Z-score transformation at each pixel
  - Build model on variations instead of raw images to ease prediction

Original

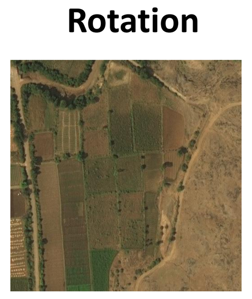
After Z-Score Standardization



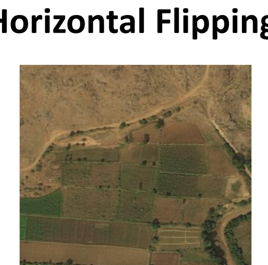
- **Data Augmentation:**
  - Model will be able to generalize to unseen images better
  - Randomly apply to input images

Brightness Transform

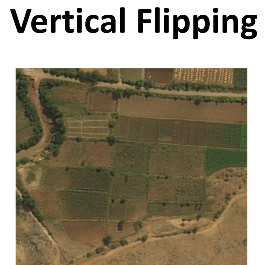
Rotation



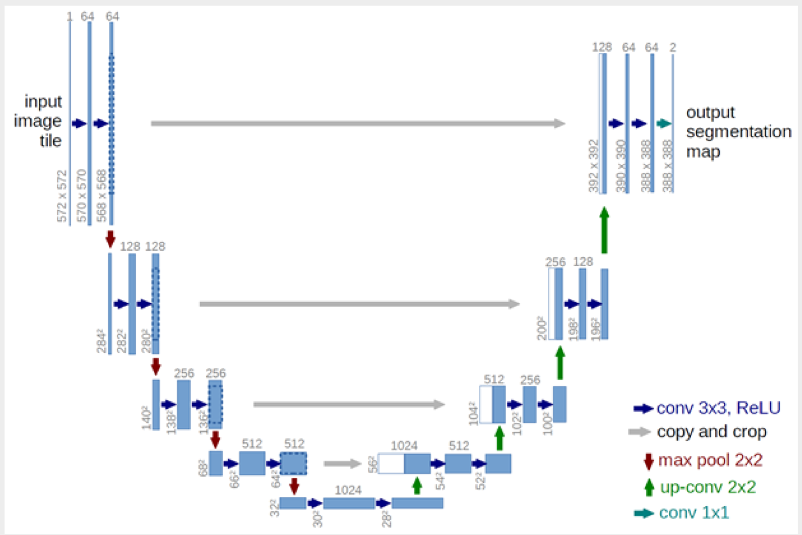
Horizontal Flipping



Vertical Flipping

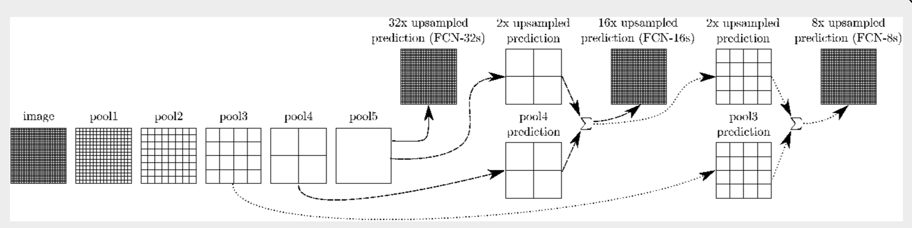


Final Model

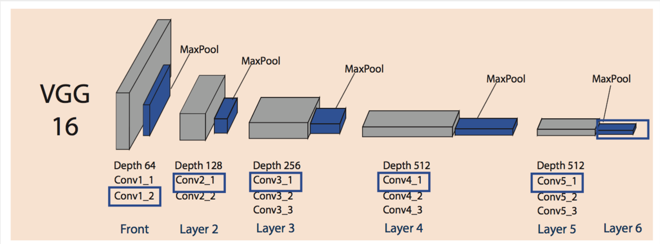


X 5

+



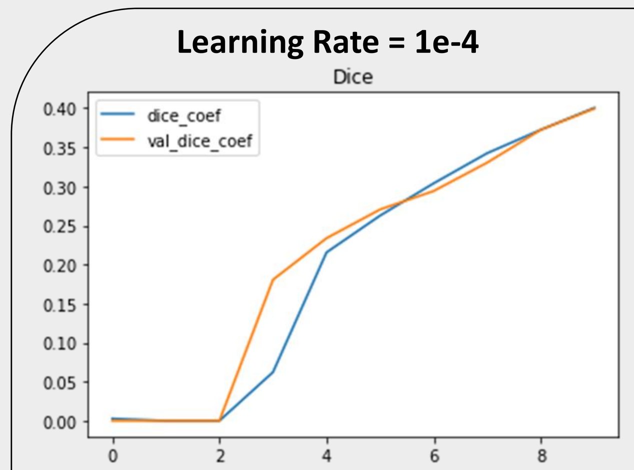
+



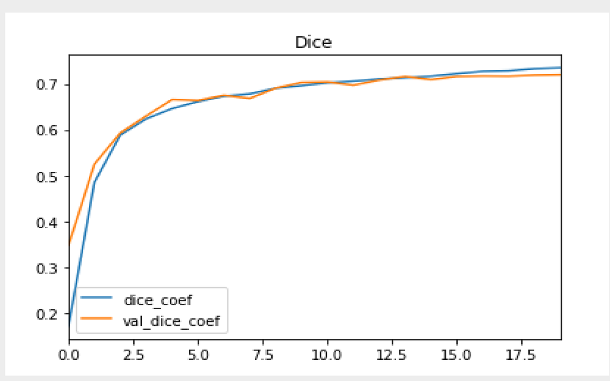
- Our final model was an ensemble of 6 convolutional neural networks consisting of 3 standard U-Nets with dropout layers, 2 modified U-Nets with batch normalization instead of dropout layers, and an FCN-8 trained starting from pre-trained VGG-16 weights
- Ensembling boosted our accuracy by around 3%
- The 3 standard U-Nets consisted of the following: 40 epochs/no augmentation, 50 epochs/no augmentation, 50 epochs/augmentation
- The 2 modified U-Nets consisted of the following: 30 epochs/augmentation, 40 epochs/augmentation

Hyperparameter Selection

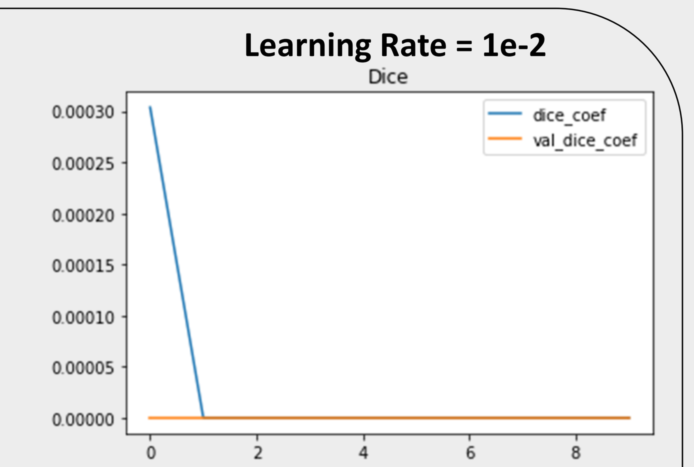
Learning Rate = 1e-4



Learning Rate = 1e-3



Learning Rate = 1e-2



- The learning rate of 1e-3 was used to train each U-Net as well as the FCN-8
- The maximum batch size which fit into memory was always used as a larger batch size typically allows for a more accurate gradient calculation (32 for the U-Nets and 1 for the FCN-8)
- The Adam optimizer was used as it is generally agreed to be the most efficient
- The training data was split as such: 90% training, 10% validation (large amount of training data is effective, and validation size still reasonable)

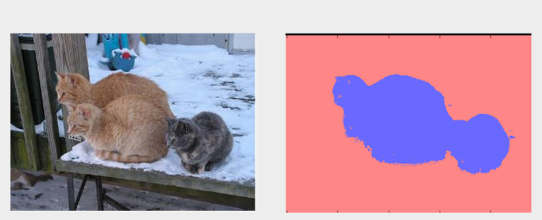
Post-Processing Steps

Conditional Random Fields (CRF)

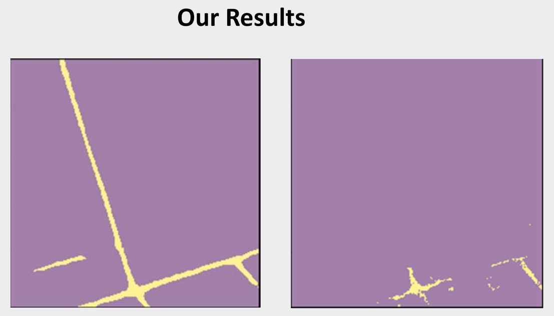
$$k(\mathbf{f}_i, \mathbf{f}_j) = w^{(1)} \exp\left(-\frac{|p_i - p_j|^2}{2\theta_\alpha^2} - \frac{|I_i - I_j|^2}{2\theta_\beta^2}\right) + w^{(2)} \exp\left(-\frac{|p_i - p_j|^2}{2\theta_\gamma^2}\right)$$

appearance kernel; smoothness kernel

Tutorial CRF Results

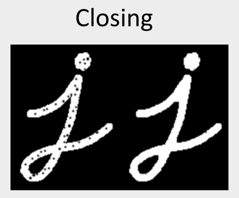


Our Results

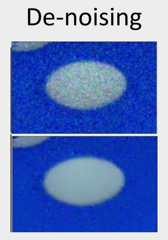


OpenCV Image Processing

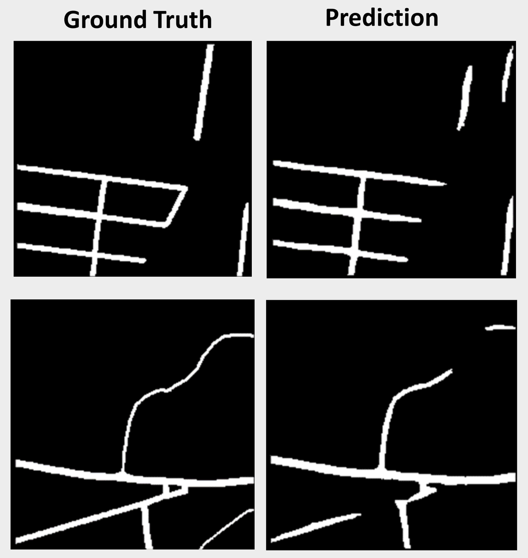
Closing



De-noising

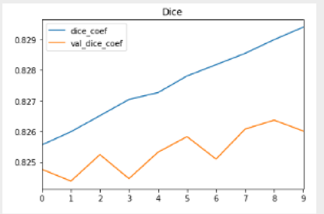


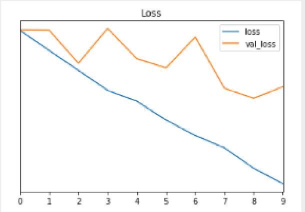
Ground Truth **Prediction**



Feeding Predicted Masks into U-Net

- The predicted masks for all the training examples generated by our top model were used as new training examples paired with the original training masks as labels
- The default U-Net was then trained on this new dataset
- Increased accuracy by around 2%





Ground Truth

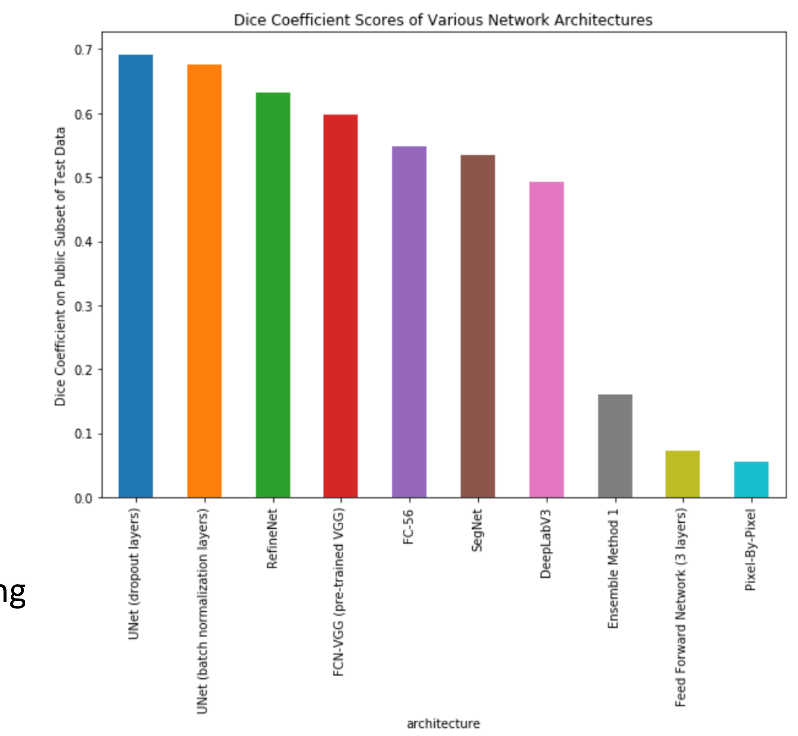
Prediction

Training & Model Selection

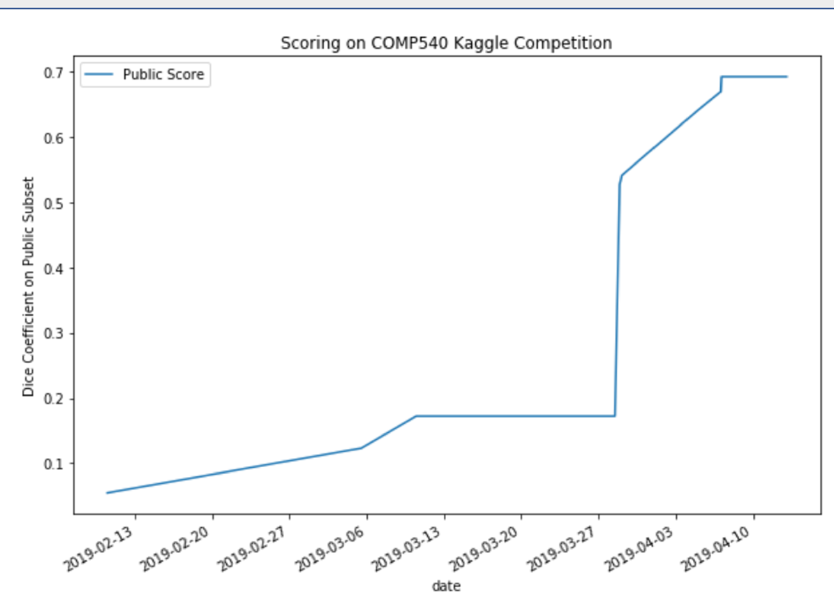
- Models trained using Amazon Web Services & Google Cloud Platform
- Regularization using Dropout, and later Batch Normalization
- Standard Train-Validation split, evaluated after every epoch
- Scoring: Dice Coefficient

$$\frac{2TP}{2TP + FP + FN}.$$

- Large dependency on recognizing road pixels



Timeline of Progress



- Massive spike when moving to Deep Convolutional Neural Networks (Late March)
- Progress slowed after tuning parameters for UNet
- Ensemble methods using majority vote and average value failed to improve

Future Directions

- Unsupervised Learning
  - Attempt to cluster types of images (country, city, etc) and build one model for each.
  - To predict on unseen images, classify based on clusters and then predict using relevant model
- Fix postprocessing using Conditional Random Fields
  - Road boundaries are almost always very smooth
  - Smooth our predictions after the fact to reflect this
- Integrated Conditional Random Fields within Neural Network Architecture
  - Instead of postprocessing, using a conditional random field as part of the network
  - Implement CRF as Recurrent Neural Network

Acknowledgements

Technical References
https://arxiv.org/abs/1703.06870
https://devblogs.nvidia.com/solving-spacenet-road-detection-challenge-deep-learning/
http://openaccess.thecvf.com/content_cvpr_2018_workshops/papers/w4/Buslaev_Fully_Convolutional_Network_CVPR_2018_paper.pdf
https://medium.com/ymedialabs-innovation/data-augmentation-techniques-in-cnn-using-tensorflow-371ae43d5be9
https://medium.com/@rogerxujiang/setting-up-a-gpu-instance-for-deep-learning-on-aws-795343e16e44
http://www.cs.toronto.edu/~fritz/absps/road_detection.pdf
https://towardsdatascience.com/dont-use-dropout-in-convolutional-networks-81486c823c16
http://warmspringwinds.github.io/tensorflow/tf-slim/2016/12/18/image-segmentation-with-tensorflow-using-cnns-and-conditional-random-fields/
https://openreview.net/forum?id=B1Yy1BxCZ
Software
https://github.com/GeorgeSeif/Semantic-Segmentation-Suite
https://github.com/matterport/Mask_RCNN
https://github.com/sadeepj/crfasrnn_keras
https://github.com/lucasb-eyer/pydensecrf
https://fairyonice.github.io/Learn-about-Fully-Convolutional-Networks-for-semantic-segmentation.html
https://www.kaggle.com/keegil/keras-u-net-starter-lb-0-277