CS230: Multispectral Satellite Imagery Feature Detection

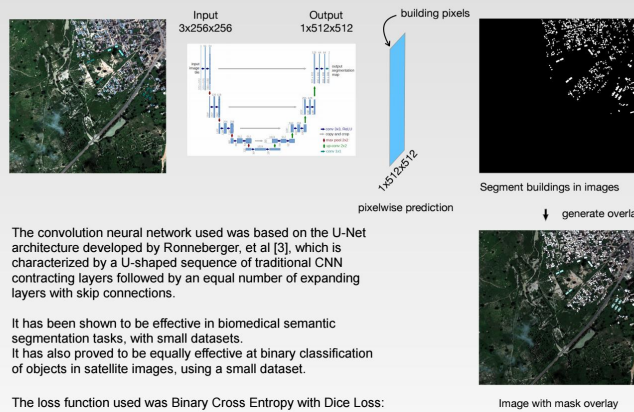Elvis Dowson <edowson@stanford.edu>

Motivation

Multispectral sensors generally capture image information across several wide-bands of the electromagnetic spectrum (typically 3 to 10). Surfaces of different objects reflect or absorb light and radiation in different ways. The ratio of reflected light to incident light is known as reflectance and is expressed as a percentage. Having multi-spectral data provides additional spectral channels to detect and classify objects in a scene based on its reflectance properties across several bands.

Copyright © 2014 Boeing. All rights reserved.
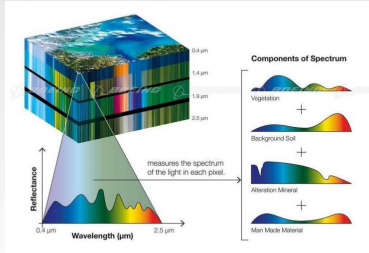
This technique can be used to detect and classify objects, additionally based on its reflectance, such as: buildings and man-made structures, roads, vegetation, water bodies, and vehicles.

Data

The dataset used for this project is based on the one provided by Kaggle for the DSTL Satellite Imagery Feature Detection (SIFD) competition [1]. It consists of a total of 25 labelled images captured across 20 different bands with varying number of channels and resolutions, e.g 3-ch RGB-band (3348 x 3403), 1-ch P-Band (3348 x 3403), 8-ch M-Band (837 x 851) and 8-ch A-Band (134 x 137).
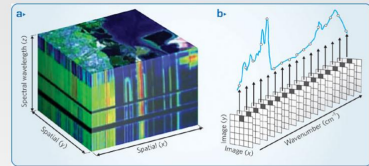
These images are raw sensor images that required a significant pre-processing such as normalization, pixel scale percentile processing, upscaling for the lower resolution bands and alignment to compensate for the way the images captured by the satellite for the different bands. The images were captured in such a way that in the M-Band, channels 2, 3, 5 and 7 were shot a few seconds after channels 1, 4, 6 and 8. This leads to ghost artifacts for moving objects, as well as a shift in the images due to the motion of the satellite over the earth.

The ground truth masks were generated using labelled polygonal shape data for 10 classes for each of the 25 images in the dataset, which includes buildings, structures, roads, tracks, trees, crops, water-ways, standing waterbodies, trucks and cars.

Neural Network Architecture

The convolution neural network used was based on the U-Net architecture developed by Ronneberger, et al [3], which is characterized by a U-shaped sequence of traditional CNN contracting layers followed by an equal number of expanding layers with skip connections.

It has been shown to be effective in biomedical semantic segmentation tasks, with small datasets.
It has also proved to be equally effective at binary classification of objects in satellite images, using a small dataset.

The loss function used was Binary Cross Entropy with Dice Loss:

$$\mathcal{L}(y_{\text{true}}, y_{\text{pred}}) = \text{binary cross entropy loss} + (1 - \text{dice co-efficient})$$

where:

binary cross entropy loss $= -\sum \left(y_{\text{true}} log(y_{\text{pred}}) + (1 - y_{\text{true}}) log(1 - y_{\text{pred}})\right)$

dice coefficient $= \dfrac{2 \sum (y_{\text{true}} \times y_{\text{pred}}) + 1}{\sum y_{\text{true}} + \sum y_{\text{pred}} + 1}$

Results

Train Image, Mask & Prediction:

Validation Image, Mask & Prediction:
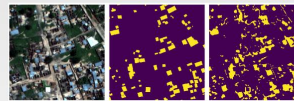
Test Image, Mask & Prediction 1:

Test Image, Mask & Prediction 2:
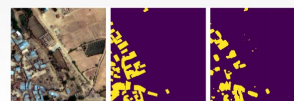
These results could potentially be improved further by applying additional post-processing to the predicted mask.

Features

The task chosen initial was that of binary semantic segmentation of buildings from satellite images.

The RGB 3-ch bands were used as input to the neural network model, in order to semantically segment buildings from the satellite images.
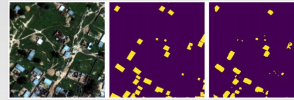
Discussions

This was a challenging project to work on, due to small size of the dataset and the amount of preprocessing required for the input images. This required data augmentation of the high-resolution images, in terms of random cropping of images and flipping.

Midway during the project, I discovered that the Python Imaging Library (PIL), used internally by PyTorch TorchVision, skimage transformer and the Augmenter library, could not handle multi-channel images and kept truncating multi-channel images and masks to 3-channels, hence only 3-channels were used with the initial U-Net model.

It took a lot of effort to get a working U-Net model with PyTorch, largely due to errors on my part, in calculating loss and accuracy metrics, due to differences in channel ordering, when dealing with Torch Tensors converted to Numpy arrays. Overall, the results were good for binary classification. The U-Net model performed well after training for 300 epochs with a batch size of 3, using 3-ch RGB images with a 512x512 input resolution.

Future Work

In terms of future work, the following tasks could be considered:

- Extend the U-Net network and evaluate its suitability for the task of multi-class image segmentation, for the 10 labelled classes in the DSTL SIFD dataset.
- Evaluate the use of a capsule network based SegCaps [4] model for binary segmentation and multi-class image segmentation; compare its performance with the U-Net architecture.
- Incorporate additional channels (e.g. Short Wave InfraRed A-Band 7, and additional reflectance indices for water and vegetation) to incorporate additional spectral information and evaluate its impact on neural network model performance.

Video Presentation

- https://youtu.be/n05b7_p4nqg

References

[1] Dstl Satellite Imagery Feature Detection
https://www.kaggle.com/c/dstl-satellite-imagery-feature-detection/data

[2] Iglovikov, Vladimir, Sergey Mushinskiy, and Vladimir Osin. "Satellite Imagery Feature Detection using Deep Convolutional Neural Network: A Kaggle Competition." arXiv preprint arXiv:1706.06169 (2017).

[3] Ronneberger, Olaf, Philipp Fischer, and Thomas Brox. "U-net: Convolutional networks for biomedical image segmentation." International Conference on Medical image computing and computer-assisted intervention. Springer, Cham, 2015.

[4] LaLonde, Rodney, and Ulas Bagci. "Capsules for Object Segmentation." arXiv preprint arXiv:1804.04241 (2018).