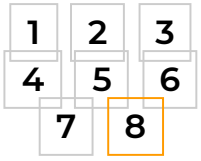
(/)

LOGOUT

OPEN HACK ENVIRONMENT OVERVIEW

OPEN HACK GUIDE PROVIDE FEEDBACK



Mark Complete

Challenge 8: The View from the Top

Background

Adventure Works has a new project for the team - building on top of the helmet detection system for safety, Adventure Works would like to identify trails that need improvements for safer travel, as another service to the community and way to advertise their brand. This will be solved with semantic segmentation.

Semantic or image segmentation is the computer vision task of assigning a label to every pixel with some set of shared characterisitcs. This segmenting can help one understand an

(/)

LOGOUT

image better by providing an easier way to view boundaries between objects or by simplifying an image to view only one type of object at a time. For instance, we may only be concerned with pixels belonging to trails in satellite images. A mask, or binary representation of this information, may be created - a mask has the same height and width of an image, but with a value (can be 1 or 0) representing presence or absence of an object of interest at that pixel.

As an example, here is a satellite image of a tract of land.

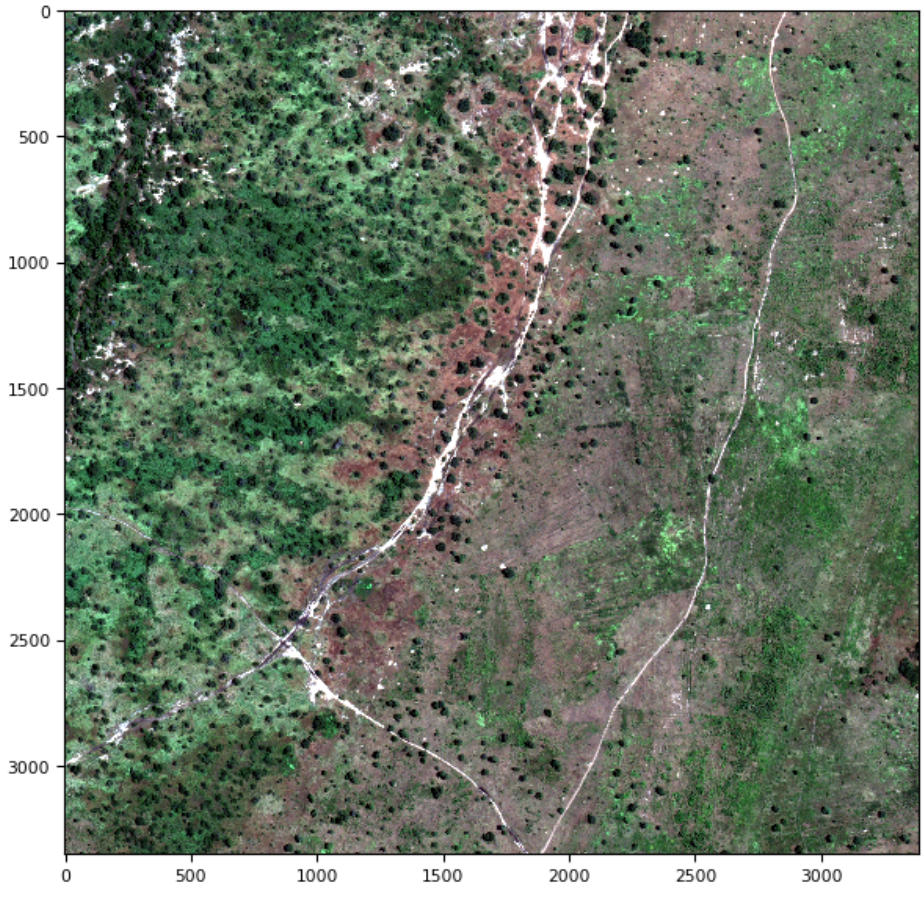
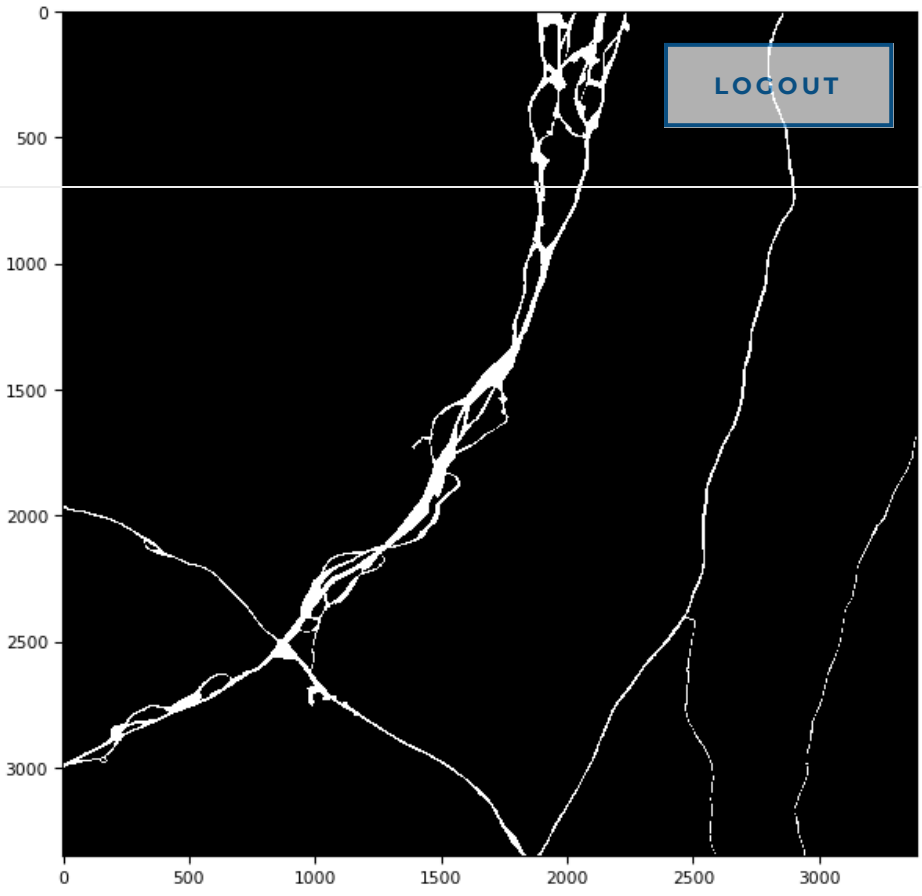


Image of Land

The mask of this image with pixels corresponding to trails is below.

(/)



Trail mask

Boundaries or contours found through semantic segmentation could be used to help surveyors decide whether or not to repair a trail based on if it looks washed out or not. This could save local governments money as they wouldn't need to send trail repair teams out unnecessarily.

In this challenge Adventure Works has asked your team to create a semantic segmentation solution.

Prerequisites

- A development environment for sharing code and working in Jupyter
- An installation of the latest version of your chosen deep learning framework(s)
- A Kaggle account - get one here (https://www.kaggle.com)

(/)

Challenge

LOGOUT

1. Download a new dataset - satellite images from Kaggle

Use the Kaggle API to download the three-band satellite images and labels. A Kaggle Account will be needed along with the account credentials.

Files needed on the shared system (extra disk storage may need to be added to VM):

- three_band.zip
- train_wkt_v4.csv
- grid_sizes.csv

IMPORTANT NOTE: The system will need to have at least 15G of space for this large dataset.

Set three images with class 4 ("Track") aside for testing the model.

Hints

- To access resources from this competition, go to https://www.kaggle.com/c/dstl-satellite-imagery-feature-detection (https://www.kaggle.com/c/dstl-satellite-imagery-feature-detection) and sign up for "Late Submission"
- Place account credentials under the `/home/<your user name>/.kaggle`
- Only images with class 4 (**Track**) should be used in training - use the information in the `csv` files to filter out the rest of the images.
- Final dimensions of the images (both input and mask) should be considered carefully. Some projects will have requirements around dimensions, e.g., the height of the

(/)

image must be strictly superior than half the width or making sure the width is even. These small notes can make a big difference, so follow the directions closely.

LOGOUT

2. Create and train a semantic segmentation model

Create a model that predicts to which class a the pixels in a set of test images belong - in this case, either trail or not trail. This can be in the form of an image mask or, simply, a set of polygon coordinates.

Hints

- The recommended architecture with which to begin is either a *UNet* or *Tiramisu* network.
- You may need to split apart the images as they are very large in this dataset.
- To rewrite an entire file in Jupyter notebook one can add `%%writefile filename.py` to the beginning of a code cell.
- Training a deep learning model is faster on a GPU machine. If the team environment does not already include GPU, consider working with your coach to adjust the environment before proceeding. The N-series virtual machine sizes in Azure support GPUs.
- Just as with classical machine learning, you should randomly split the data into training and validation subsets with which to train and validate the model. If more images are needed as training data, consider cutting the images into pieces.
- As you iterate on the model architecture, save the best trained models to disk.
- Try to avoid *overfitting* your model to the training data. One sign of this is that after your training and validation loss

(/)

metrics converge, the training loss continues to drop but your validation loss stays the same or rises.

LOGOUT

- See **References** below for resources and more information.

3. Use your model with new data

Use your model to predict the masks of 3 test images. Plot the masks as images.

Hints

- If your model output is a polygon, `shapely` library along with `opencv` may be used to convert to a mask.
- You may need to split apart the images as they are very large in this dataset

Success Criteria

- Successfully train a semantic segmentation model
- Achieve Jaccard score of **0.3** (30%) or greater using your test data set.
- Show predictions for the three images in the test data.

References

Concepts

- Detection and Segmentation - CS231n Lecture (https://www.youtube.com/watch?v=nDPWywWRIRo) (Video)
- UNet Paper (https://arxiv.org/abs/1505.04597)
- Tiramisu Paper (https://arxiv.org/abs/1611.09326)

(/)

- Evaluating image segmentation models (https://www.jeremyjordan.me/evaluating-image-segmentation-models/)

LOGOUT

Mananging Data

- Kaggle API (https://github.com/Kaggle/kaggle-api)
- Attach a data disk on Azure (https://docs.microsoft.com/en-us/azure/virtual-machines/linux/attach-disk-portal)

Tools and Frameworks

- UNet in PyTorch (https://github.com/milesial/Pytorch-UNet)
- UNet in Keras (https://www.kaggle.com/drn01z3/end-to-end-baseline-with-u-net-keras)
- UNet in TensorFlow (https://github.com/jakeret/tf_unet)

Image Libraries

- Shapely (https://shapely.readthedocs.io/en/latest/)

Code Samples

- Kaggle notebook on calculating Jaccard score and polygon masks (https://www.kaggle.com/iglovikov/jaccard-polygons-polygons-mask-polygons)
- Kaggle notebook on polygon masks and basic segmentation (https://www.kaggle.com/lopuhin/full-pipeline-demo-poly-pixels-ml-poly)

(/)

LOGOUT