

An aerial photograph showing a large, deep landslide in a rural landscape. The landslide is a prominent, reddish-brown scar cutting through green fields and vegetation. In the background, there are small houses, a pond, and more greenery. The foreground shows the edge of the landslide with exposed soil and some debris.

# Using Machine Learning to Identify Landslides

---

Nathalie Redick

McGill University



---

# A Quick Introduction

---

- I am a 4<sup>th</sup> year undergraduate at McGill University
- Majoring in computer science and minoring in earth & planetary sciences
- Primarily interested in how machine learning can be applied to geoscientific problems





# Background

What data & what model are we using?

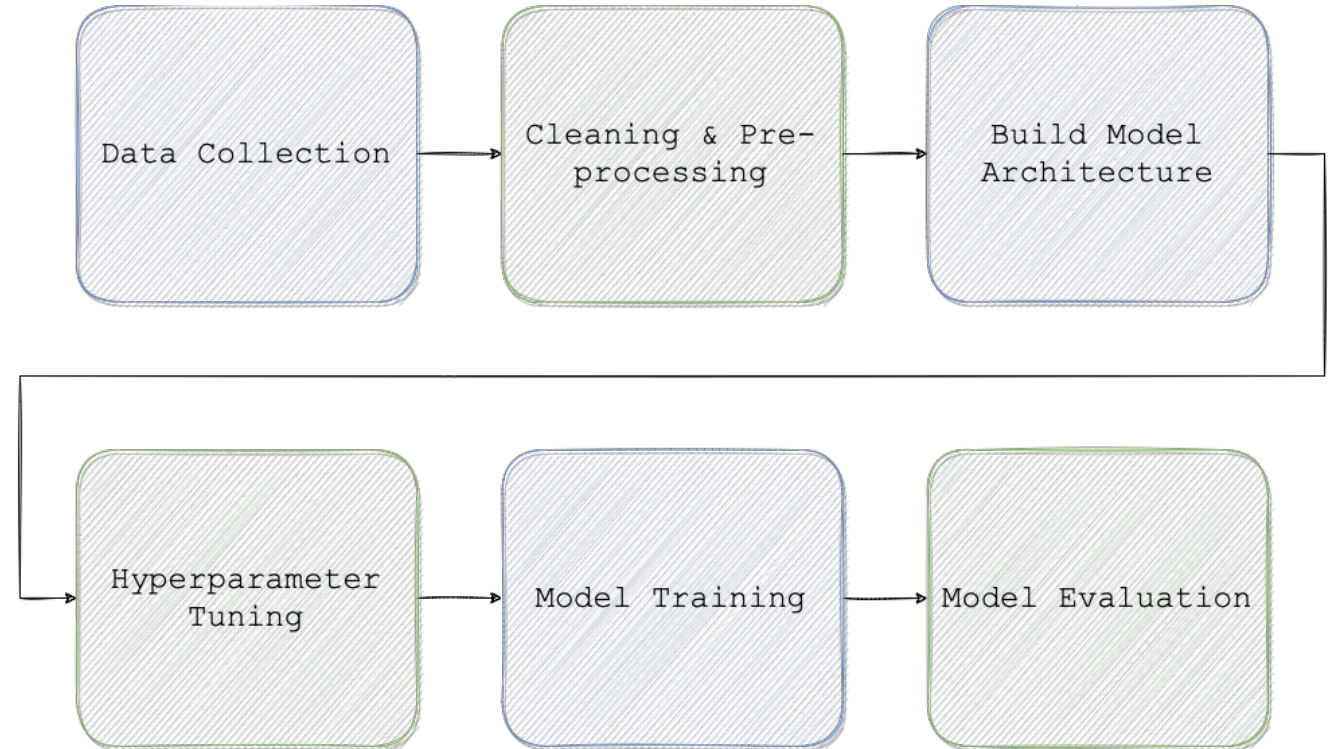
# Objectives

- Automate routines for downloading and processing publicly available input data from APIs (such as DEMs)
- Create a modular, scalable machine learning model based on the downloaded input data for detecting landslides



# Workflow

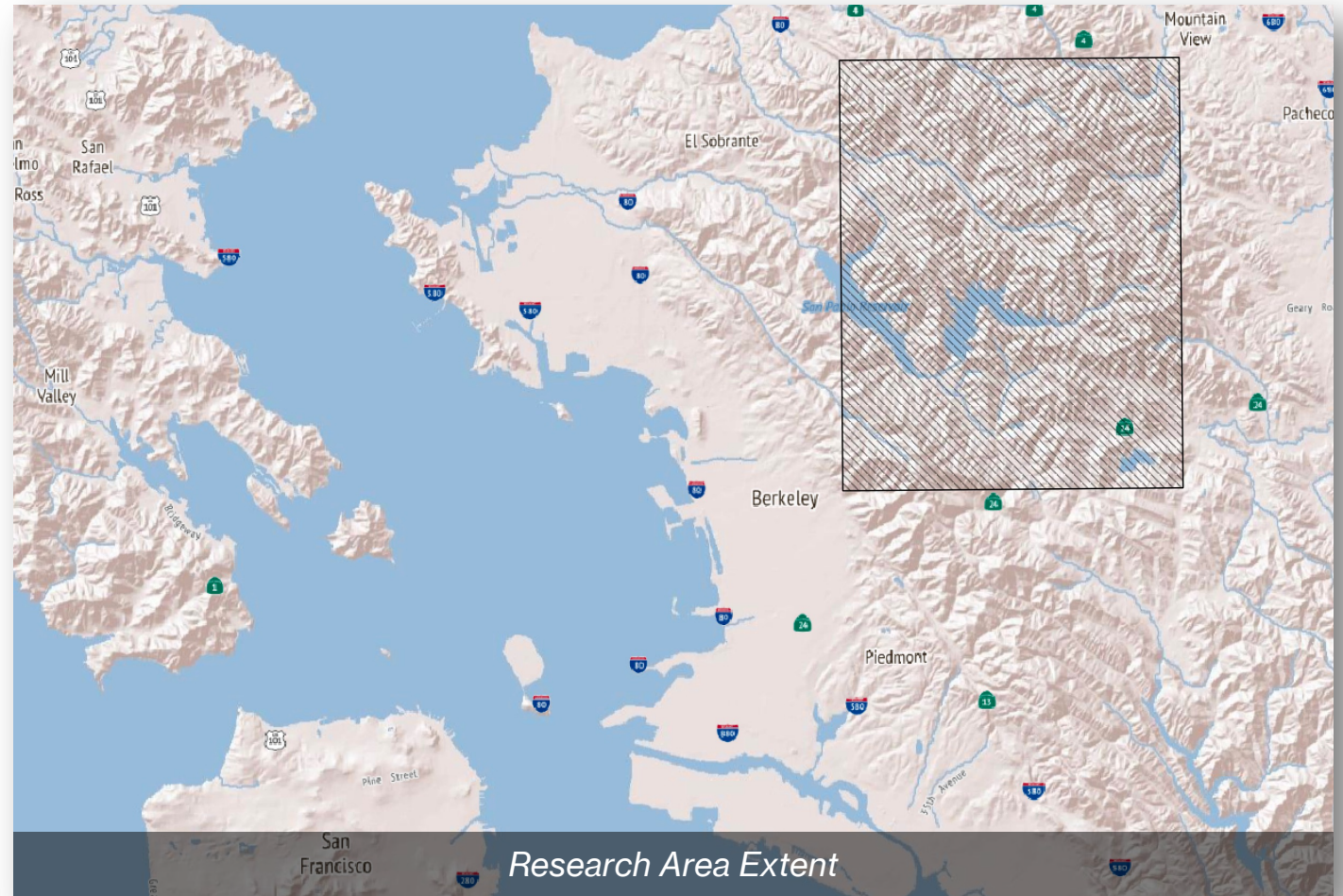
---





# Data

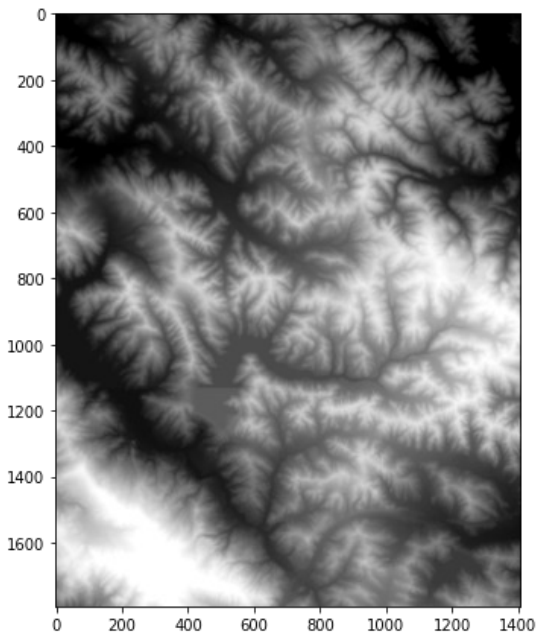
- To train a model, we need images (image tiles) and truth values for the pixels in those tiles (masks/segmentation maps)
- The Contra Costa County Landslide Inventory serve as our masks: polygons of landslide scarps and deposits
- The features are derived from a ~10m resolution DEM, retrieved from the USGS [3D Elevation Program](#) database



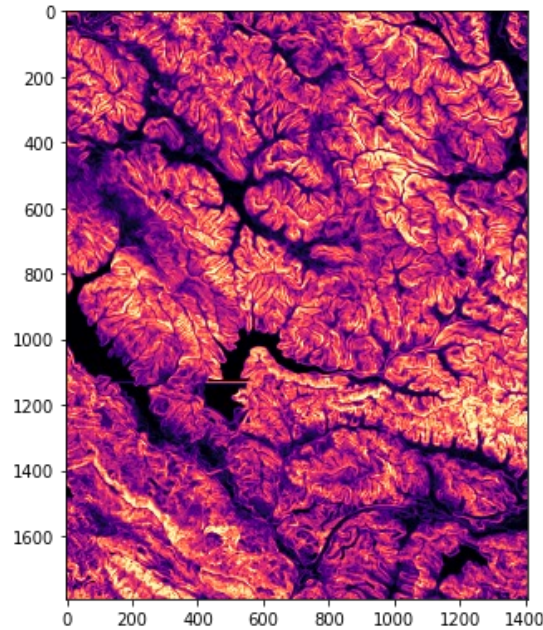
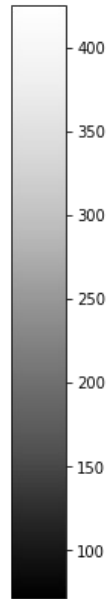


# Training Images

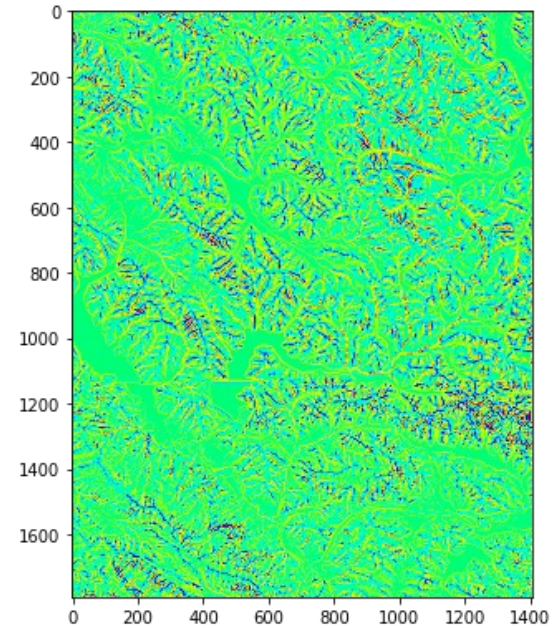
- Our training images are a composite raster created from DEM derivatives of the Contra Costa quadrangle
- Each layer becomes a channel in the image, like RGB channels



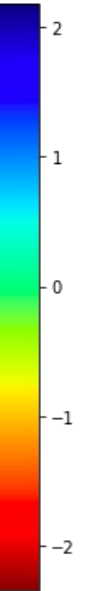
*Elevation*



*Slope*

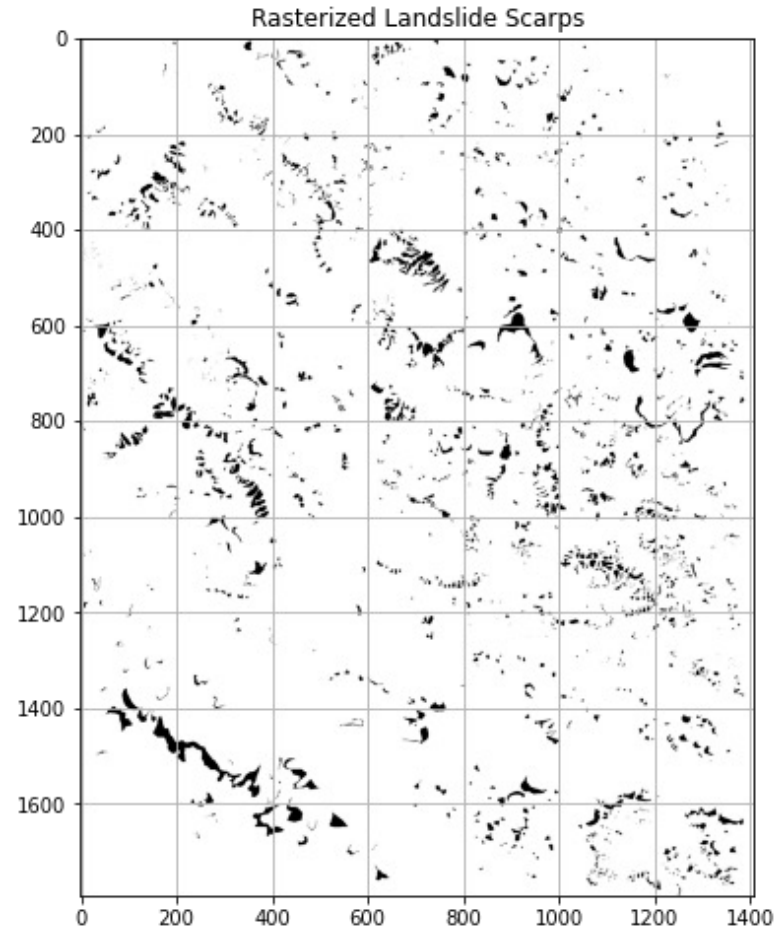
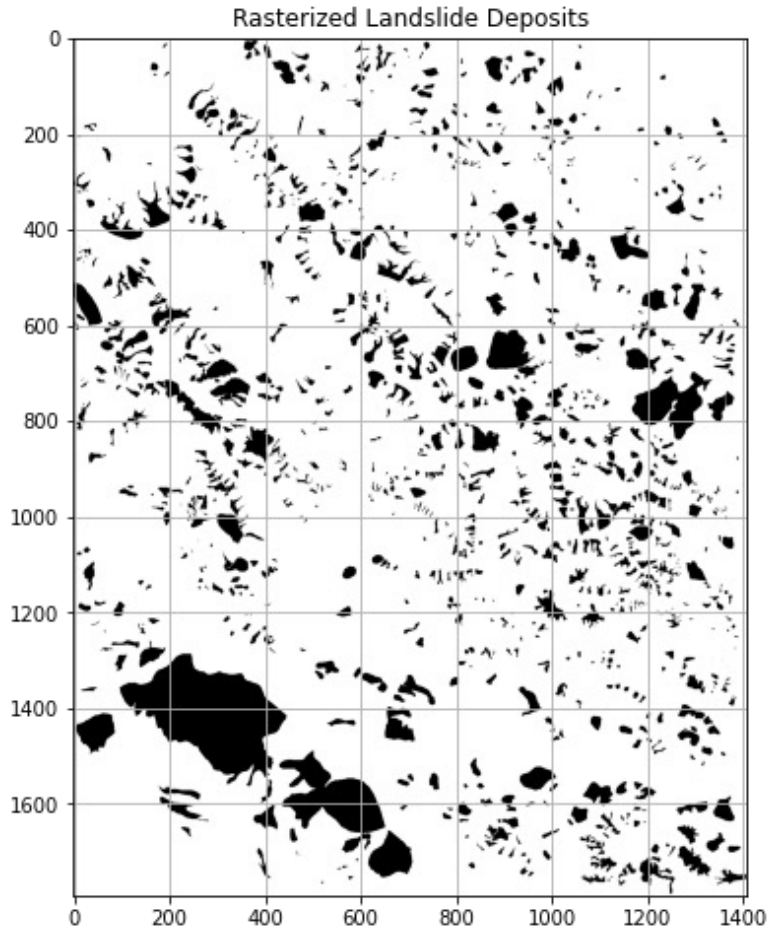


*Curvature*





# Feature Masks

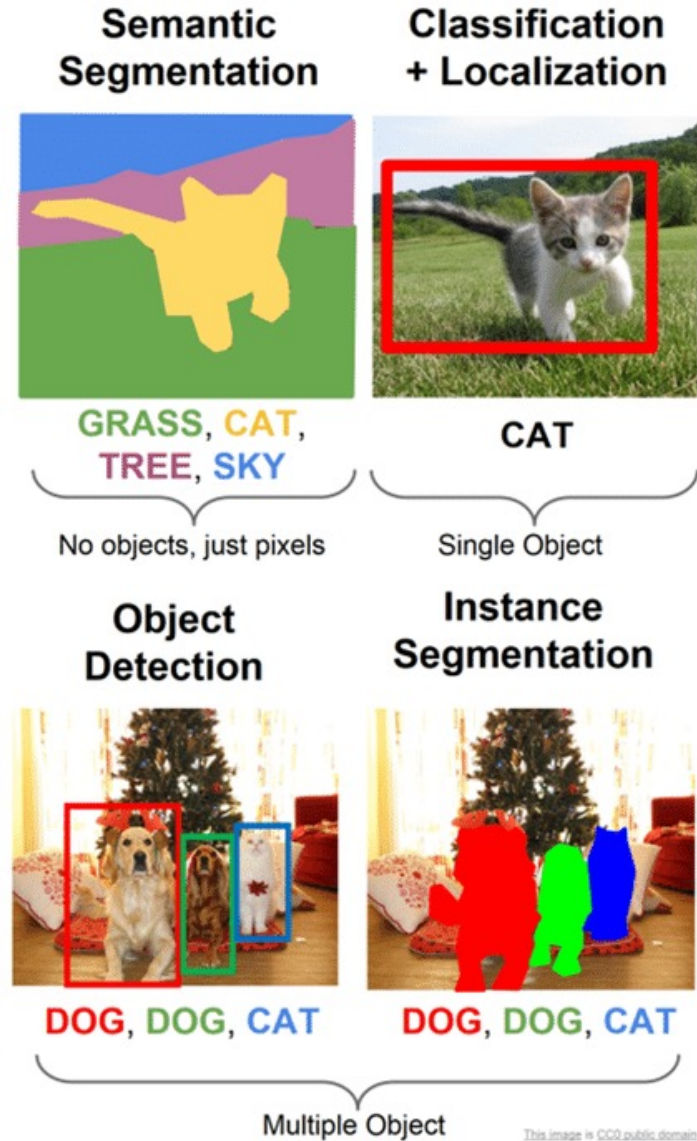


- The masks are produced from the landslide scarp and deposit rasters
- The polygons are rasterized to the same resolution as the composite image (approximately 10m per pixel)



# U-Net Model

- A U-Net is a convolutional neural network (CNN) designed for semantic image segmentation, named for its “U” shape
- It makes predictions on a pixel-wise basis
- The model has two paths: **down-sampling** (left side) and **up-sampling** (right side)
  - Down-sampling extracts image features
  - Up-sampling localizes objects (landslides)
- This kind of neural network is well-suited to our task because it can preserve spatial relationships in the data
- Neural networks can perform well when you have limited data



Source: [Li, Johnson & Yeung \(2017\)](#)

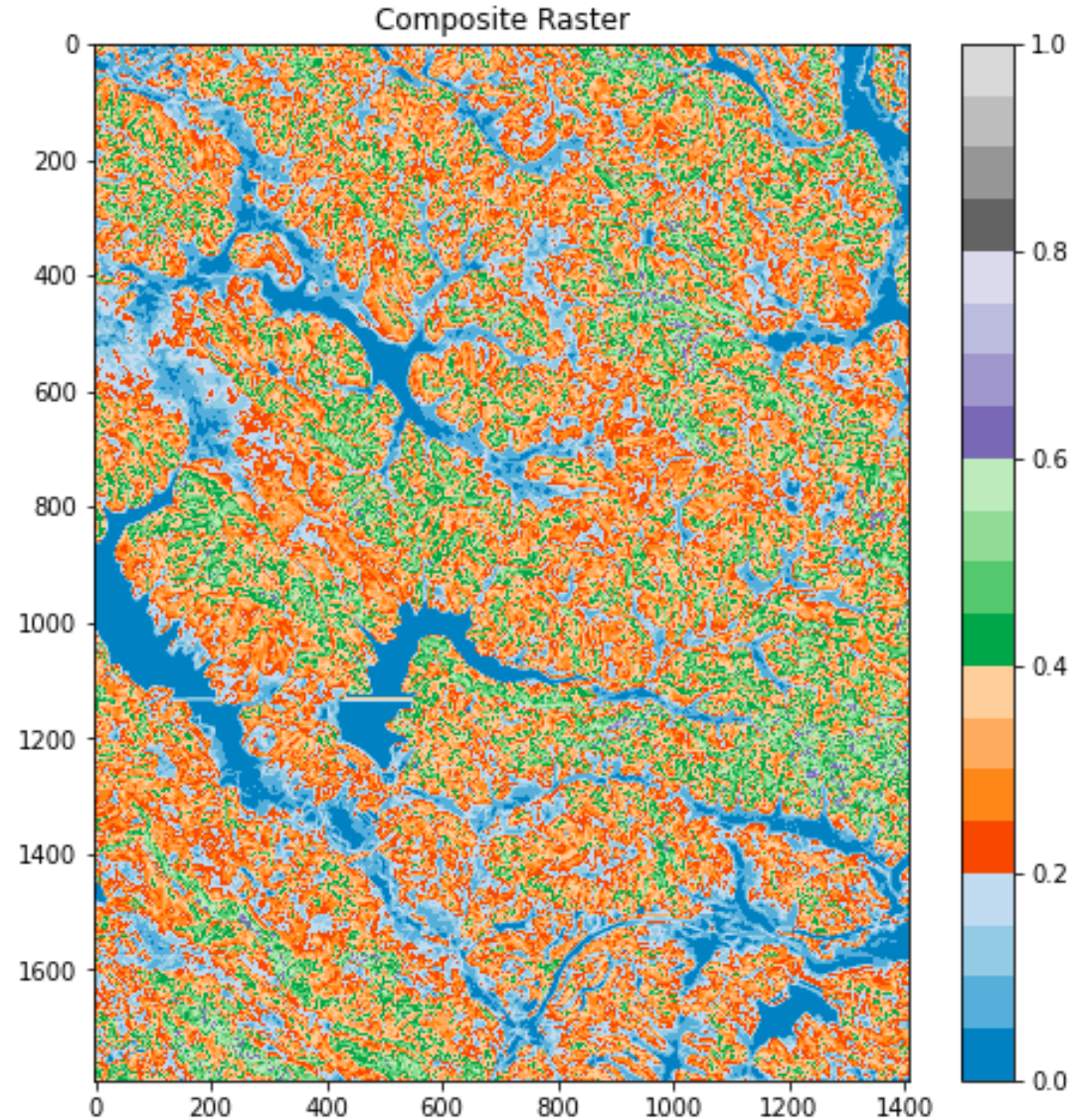


# Data Preprocessing

How are we processing the data & why?

# Rasterization

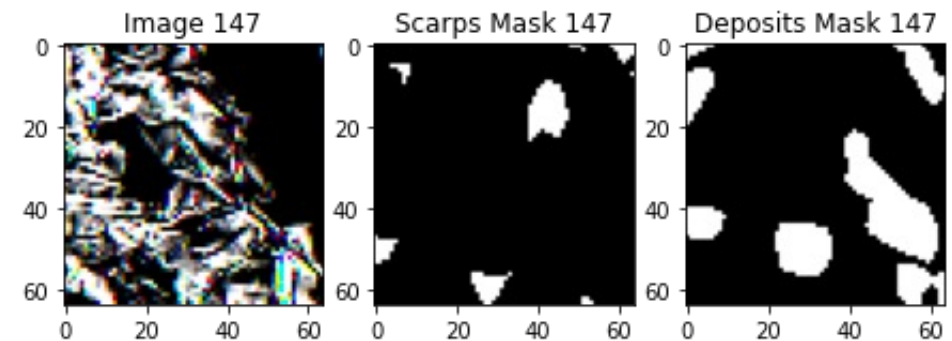
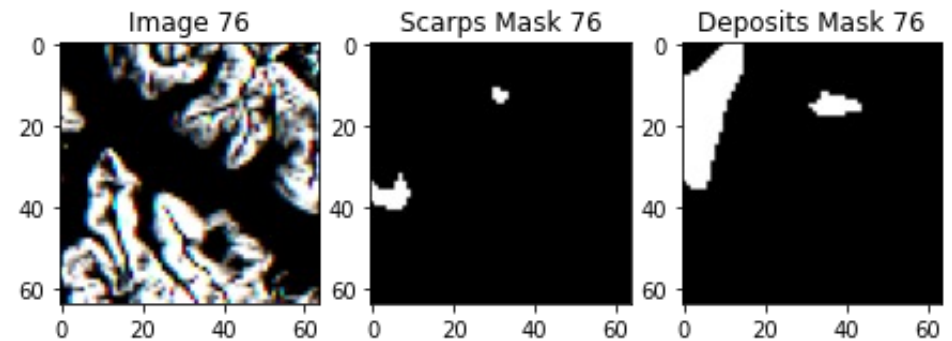
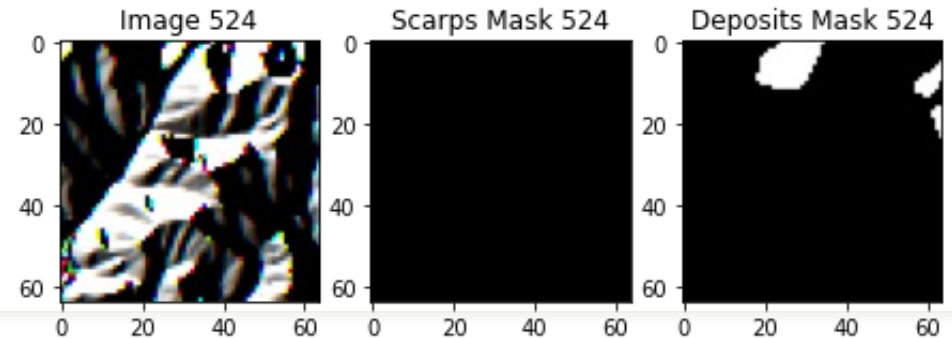
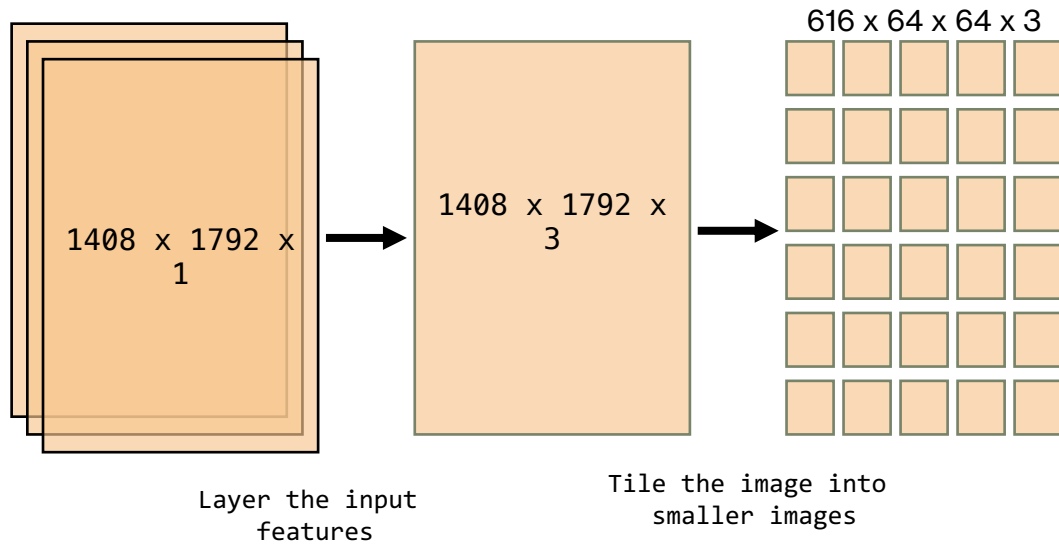
- After the DEM is downloaded from the 3DEP server, we derive slope & curvature using the richdem library
- These 3 features are combined into a composite raster
  - Note that the values have been normalized to  $[0, 1]$  for better model performance
- The scarp & deposit polygons are rasterized into binary **masks**





# Tiling the images

- The images are tiled to 64 by 64 pixels
- These are our training images (616 tiles) & aligned feature masks (616 scarp masks & 616 deposit masks)



Example training images & masks

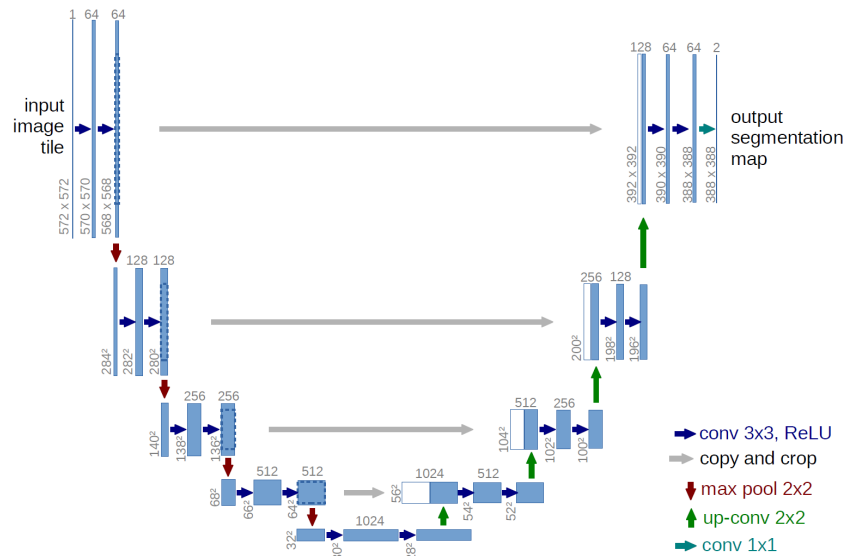
# Data Augmentation

- 616 tiles is not a lot of data for the model to be able to learn patterns
- We resolve this issue by combining the original 616 tiles with *augmented* tiles
- The augmented tiles are created using the `imgaug` library
  - 50% chance of an image being flipped horizontally
  - 50% chance of an image being flipped vertically
- This allows us to double the size of the training dataset (1232 tiles)
- It allows helps the model generalize better when it sees new data



# Training the U-Net Model

Defining & training a model to fit the data



Source: [University of Freiburg](https://www.unifreiburg.de)

# Building the Model Architecture

- Features are extracted from the tiles via down-sampling in the left half of the “U”
  - Captures **context**
  - Composed of a stack of convolution and max pooling layers
- Predicted regions (landslide/non-landslide) are localized in the right half of the “U”
  - Composed of transposed convolutional layer(s)
- U-Net is a kind of fully-connected neural network

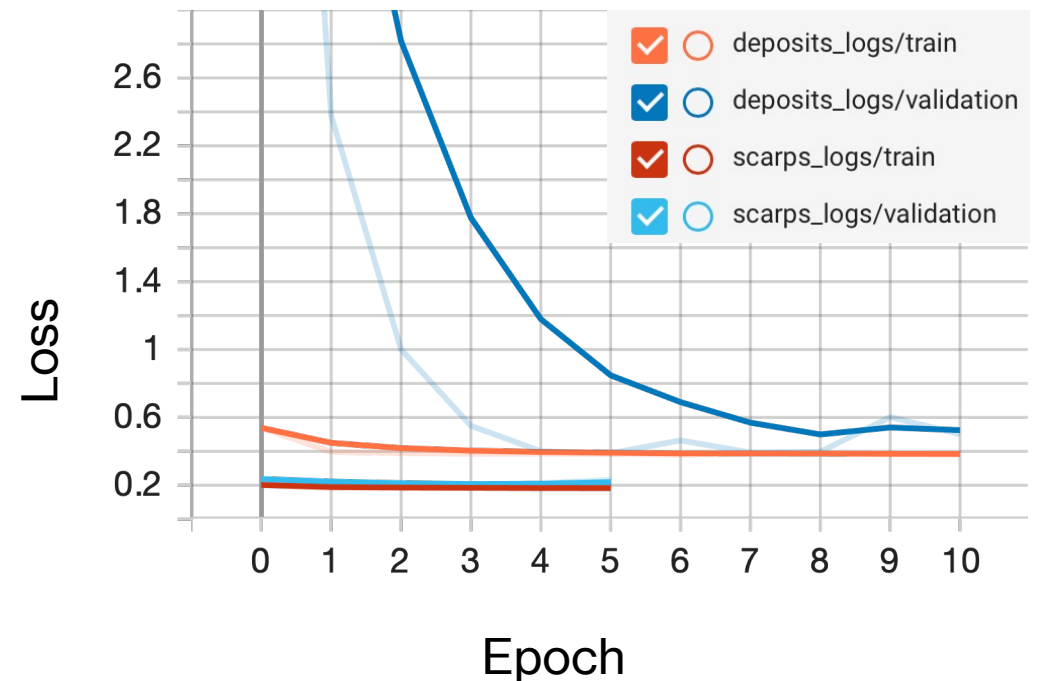
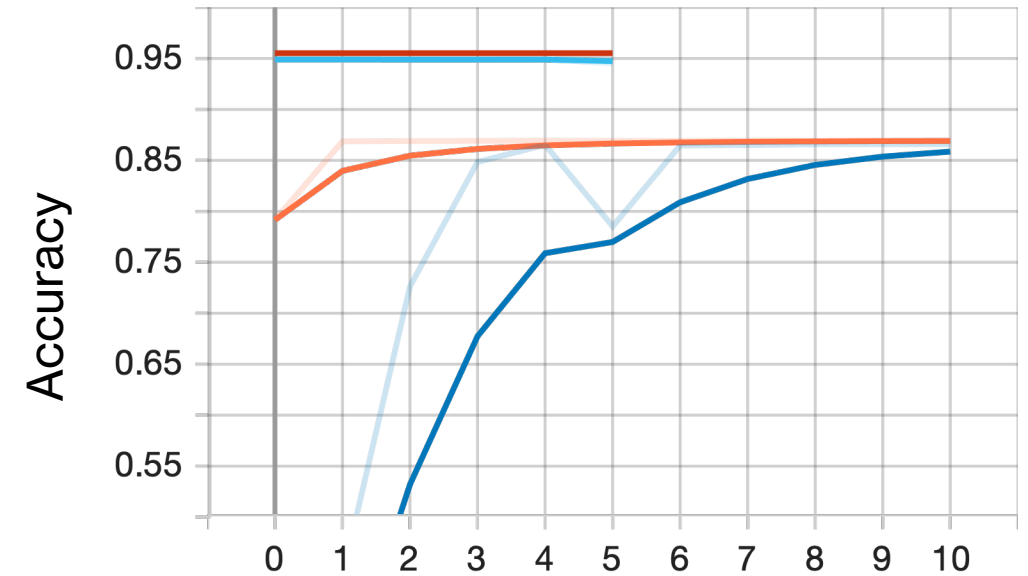


# Defining Hyperparameters

- Hyperparameters determine the parameters that the model learns by controlling the learning process
- Some of the hyperparameters used to tune this model include input tile size, batch size, optimization function, loss function, activation function, and train/valid/test split
  - Optimization function (Adam) guides the model towards a lower loss, where loss is calculated using binary cross-entropy
  - Activation function (sigmoid) defines the output segmentation mask
  - These particular functions are best geared toward a neural network training for a binary semantic segmentation problem

# Training the Model

- Two models are trained using the U-Net architecture
  - One to isolate landslide scarps & one to isolate landslide deposits
- The model trains over iterations called **epochs**
- It continues to train until the loss has stopped improving for two consecutive epochs (early stopping)
- Accuracy and loss are monitored over the course of training
  - Training loss & validation loss per epoch





# Side note

- The training metrics are uploaded to an [interactive Tensorboard](#) that allows them to be viewed publicly
- Also able to see details about training times and specific performance metric values per epoch

The screenshot displays the TensorBoard interface for the 'Contra Costa Landslides U-Net Model'. The top navigation bar includes links for SCALARS, GRAPHS, HISTOGRAMS, DISTRIBUTIONS, HPARAMS, and TEXT. The main content area shows training metrics for 'epoch\_accuracy' and 'evaluation\_accuracy\_vs\_iterations'. A table overlay provides detailed data for the 'epoch\_accuracy' metric.

Name	Smoothed	Value	Step	Time	Relative
deposits_logs/train	0.8647	0.8693	4	Mon Sep 12, 20:25:22	1h 21m 9s
deposits_logs/validation	0.7589	0.8653	4	Mon Sep 12, 20:25:22	1h 21m 9s
scarps_logs/train	0.9552	0.9552	4	Mon Sep 12, 20:49:08	35m 5s
scarps_logs/validation	0.9489	0.949	4	Mon Sep 12, 20:16:57	2m 53s

# Results

How do you determine model performance & how well did our model do?



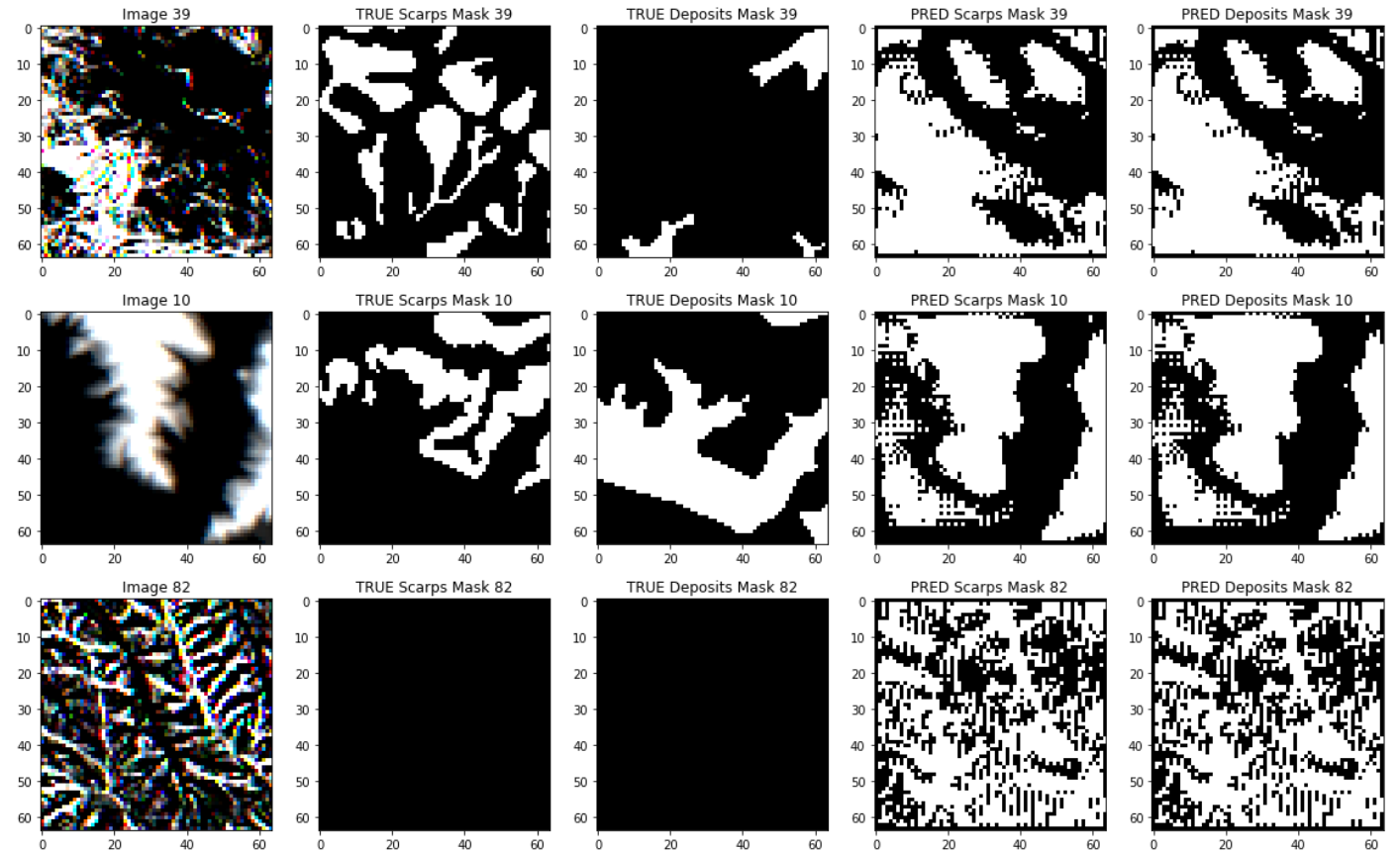
# Performance Metrics

- At this point in time, we are only looking at two performance metrics: loss and accuracy
  - Loss is the difference between the expected outcome and the model's predicted outcome
  - Accuracy is the number of correctly predicted data points out of all of the data points
- Both metrics are given as values in the range [0, 1]
  - In short, accuracy should approach to 1 and loss should approach 0
- We look at the **test** loss and accuracy values since the model has never seen those before
  - These test values are calculated from the trained model on a test set

Model	Test Loss	Test Accuracy
Deposits U-Net	0.410	0.861
Scarps U-Net	0.198	0.953

# Visualizing the Results

- We can see that the predicted masks look more like the composite than the true masks
- Despite the high accuracy values, the model is not producing great outputs
- This is a common result of having too little data to train with



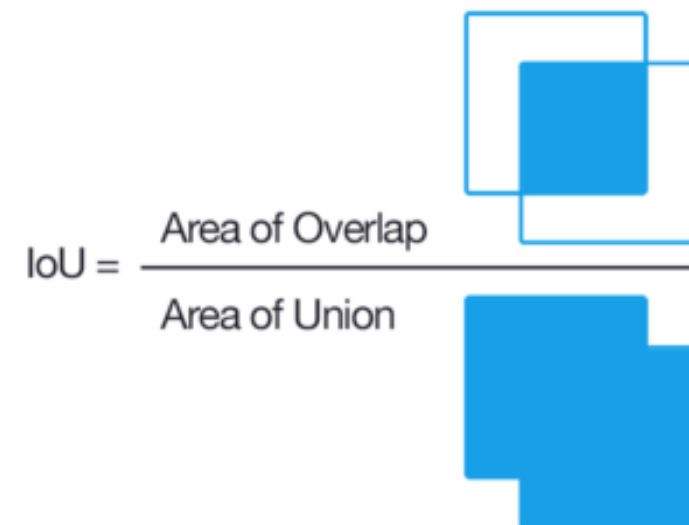


# Next Steps

Fixing current issues & thinking about the future

# Remedying Overfitting

- The best (and easiest) way to improve a model that is overfitting is to acquire more training data
- Add more **regularization** to the model
  - Regularization techniques help the model stay in line
  - For our purposes, adding some dropout layers may help the model generalize better and prevent it from memorizing the input composite
- Use more performance metrics to better understand how the model is learning (and where it may be going wrong)
  - Precision, recall & IoU


$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$

Source: [Wikipedia](#)

# Taking it a Step Further

- It would be interesting to see how this technique compares to current techniques, such as spectral methods
- Incorporate input features such as soil, lithology, climate, and vegetation
  - Use backpropagation techniques to understand what the model considers to be most important when trying to find a landslide scarp/deposit



# Significance

- While the output masks still leave something to be desired, the automated workflow for downloading input data, processing it, and training a model
  - The model isn't landslide-specific!
- The workflow is portable and scalable



# Thank you!

---

Special thanks to Drs. Jamie Kirkpatrick, Veronica Prush, Matthew Tarling, & Jin Guo for their guidance & patience.