

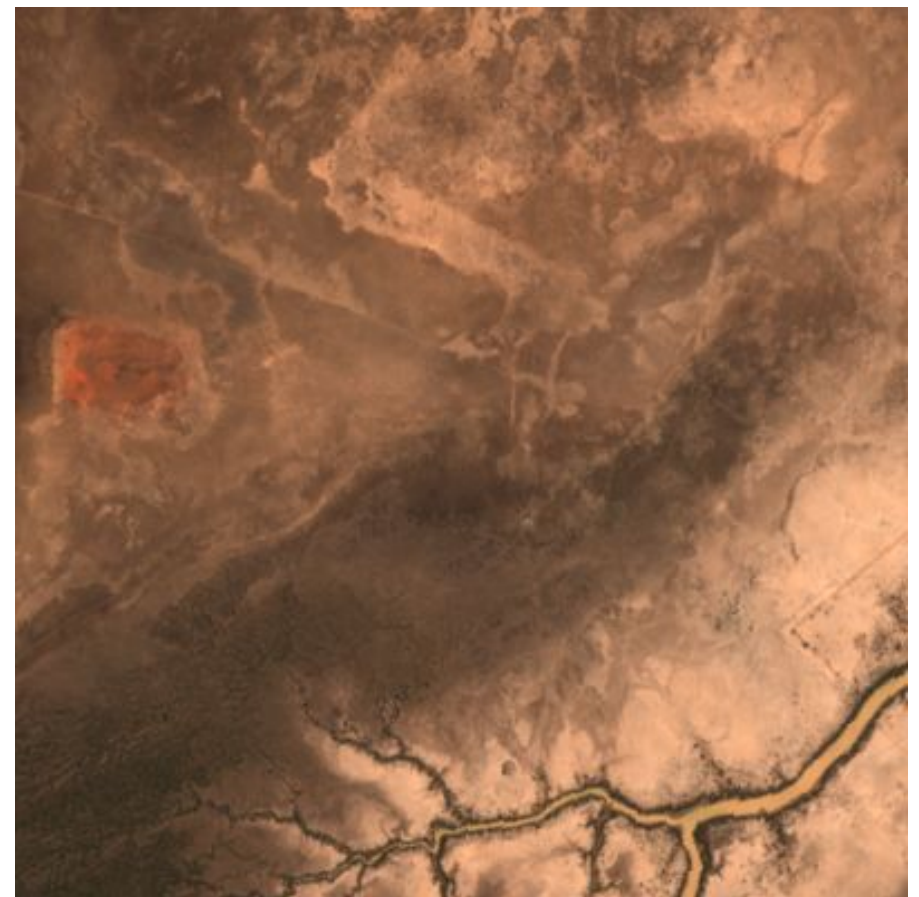
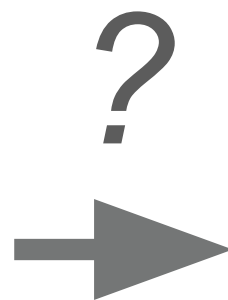
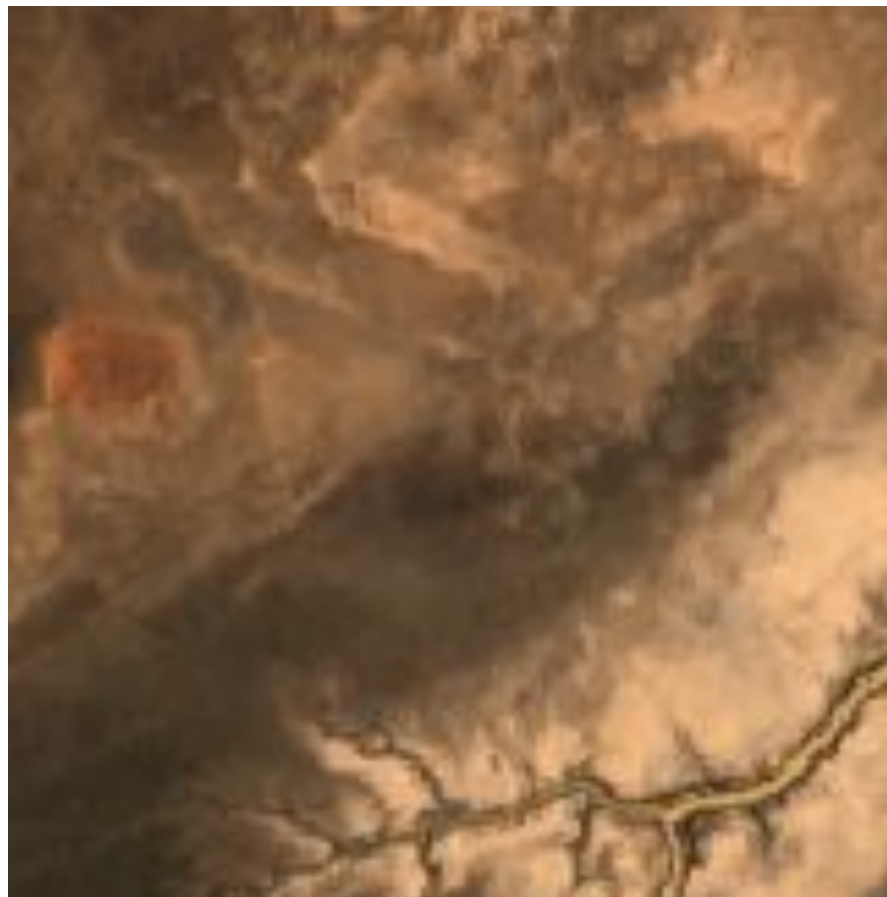
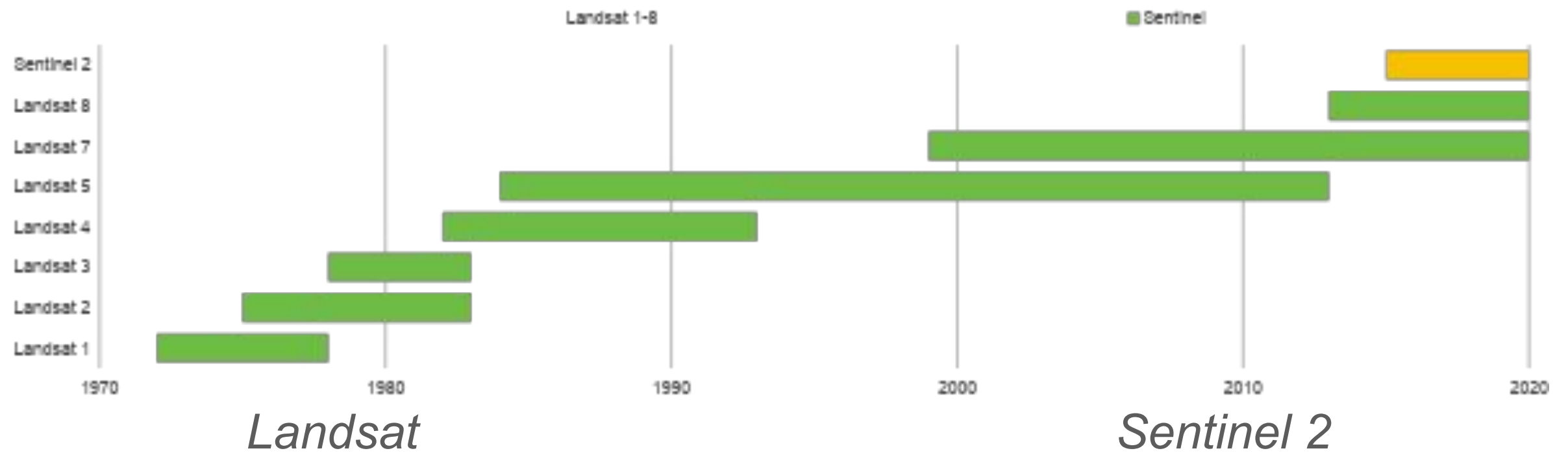
SUPERSAT

*Boosting remotely sensed data
with deep learning*

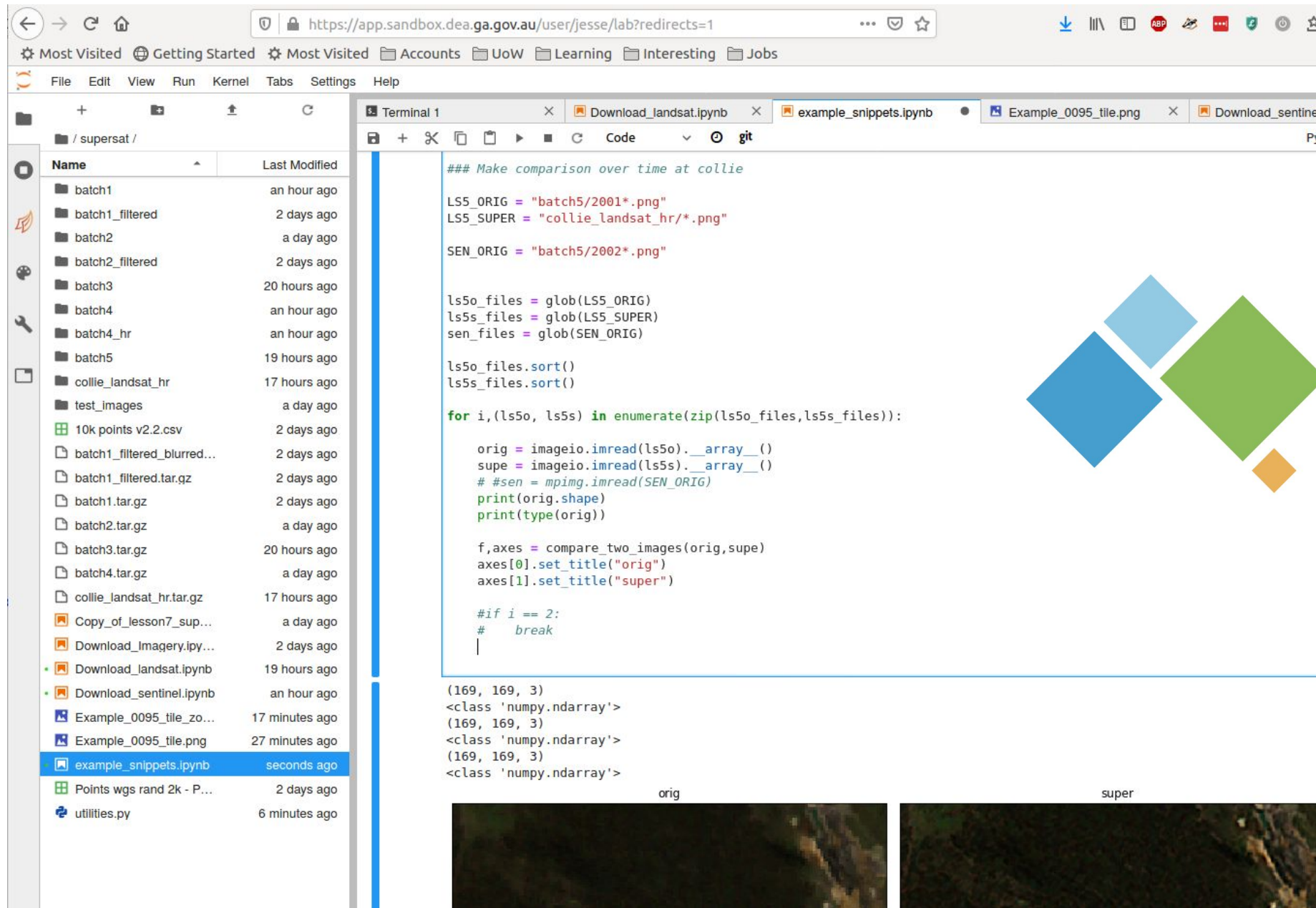
Jesse Greenslade

Nick Wright

PROBLEM



GETTING THE DATA



The screenshot shows a web-based JupyterLab interface. The left sidebar displays a file explorer for the directory `/supersat/`, listing various satellite image batches and processed files. The main area contains a code editor with a Python script for image comparison and a terminal window showing the execution output.

File Explorer:

Name	Last Modified
batch1	an hour ago
batch1_filtered	2 days ago
batch2	a day ago
batch2_filtered	2 days ago
batch3	20 hours ago
batch4	an hour ago
batch4_hr	an hour ago
batch5	19 hours ago
collie Landsat_hr	17 hours ago
test_images	a day ago
10k points v2.2.csv	2 days ago
batch1_filtered_blurred...	2 days ago
batch1_filtered.tar.gz	2 days ago
batch1.tar.gz	2 days ago
batch2.tar.gz	a day ago
batch3.tar.gz	20 hours ago
batch4.tar.gz	a day ago
collie Landsat_hr.tar.gz	17 hours ago
Copy_of_lesson7_sup...	a day ago
Download_imagery.ipynb	2 days ago
Download_Landsat.ipynb	19 hours ago
Download_sentinel.ipynb	an hour ago
Example_0095_tile_zo...	17 minutes ago
Example_0095_tile.png	27 minutes ago
example_snippets.ipynb	seconds ago
Points wgs rand 2k - P...	2 days ago
utilities.py	6 minutes ago

Code Editor:

```
### Make comparison over time at collie

LS5_ORIG = "batch5/2001*.png"
LS5_SUPER = "collie_Landsat_hr/*.png"

SEN_ORIG = "batch5/2002*.png"

ls5o_files = glob(LS5_ORIG)
ls5s_files = glob(LS5_SUPER)
sen_files = glob(SEN_ORIG)

ls5o_files.sort()
ls5s_files.sort()

for i, (ls5o, ls5s) in enumerate(zip(ls5o_files, ls5s_files)):

    orig = imageio.imread(ls5o).__array__()
    supe = imageio.imread(ls5s).__array__()
    # sen = mpimg.imread(SEN_ORIG)
    print(orig.shape)
    print(type(orig))

    f, axes = compare_two_images(orig, supe)
    axes[0].set_title("orig")
    axes[1].set_title("super")

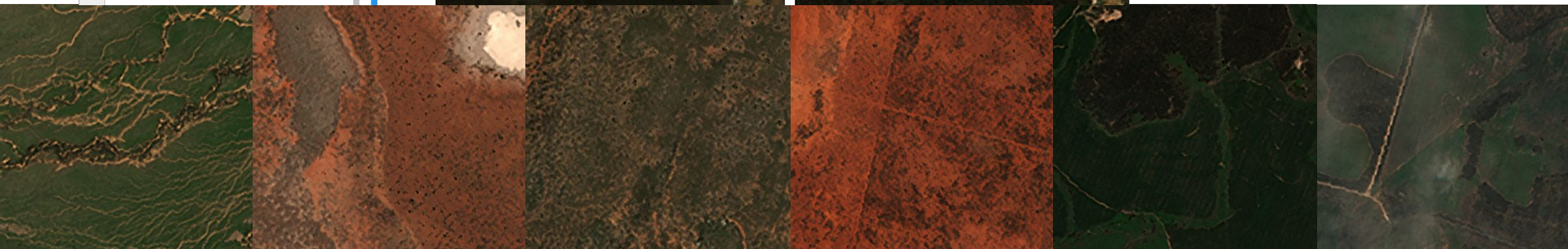
    #if i == 2:
    #    break
    |
```

Terminal Output:

```
(169, 169, 3)
<class 'numpy.ndarray'>
(169, 169, 3)
<class 'numpy.ndarray'>
(169, 169, 3)
<class 'numpy.ndarray'>
```

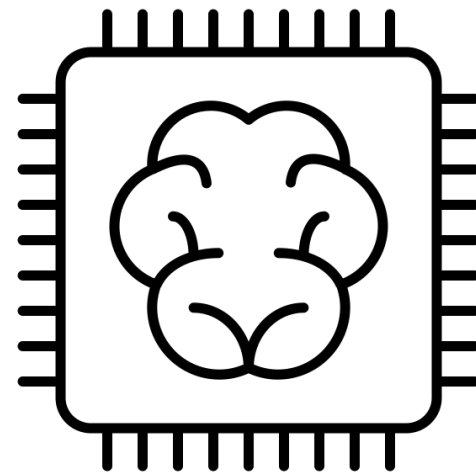


Digital Earth
AUSTRALIA

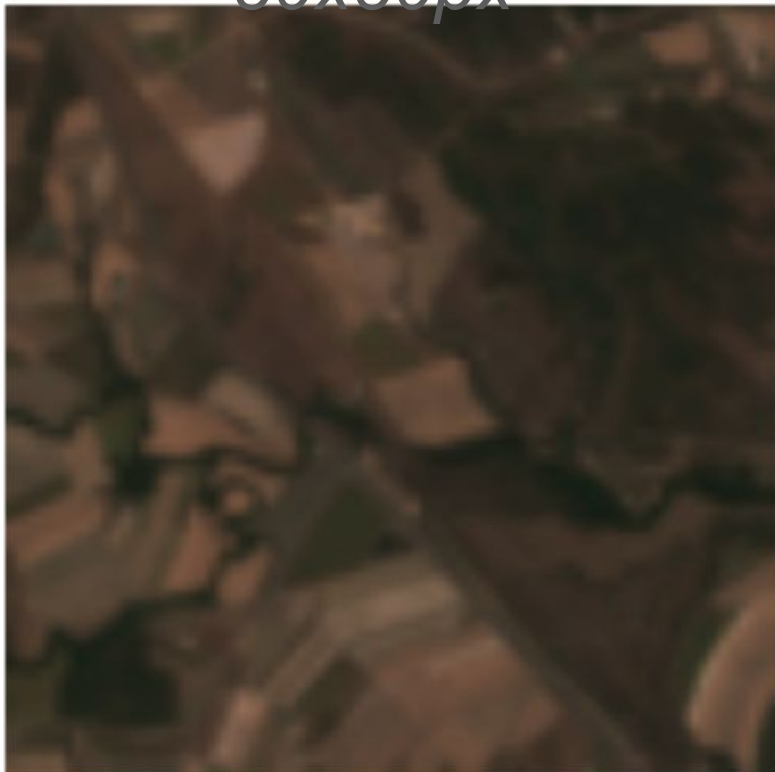


TRAINING THE MODEL

*Fastai library on Google
Colab UNET, VGG,
ResNet34, pertained on
ImageNet*



*Input
Sentinel
80x80px*



*Prediction
Sentinel
400x400px*

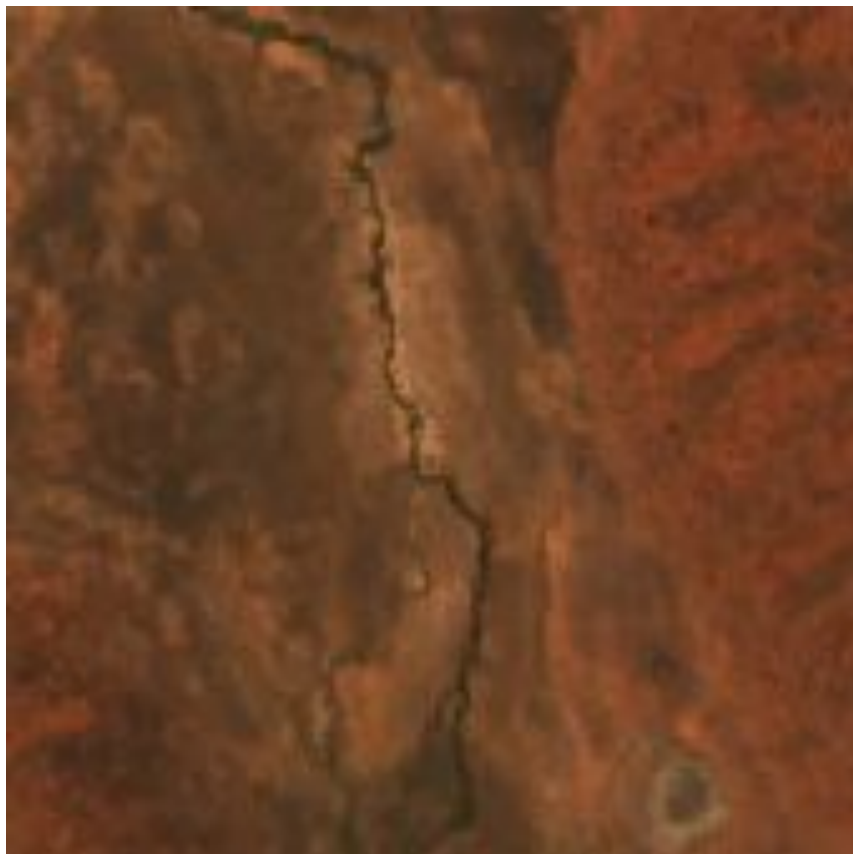


*Target Sentinel
400x400px*

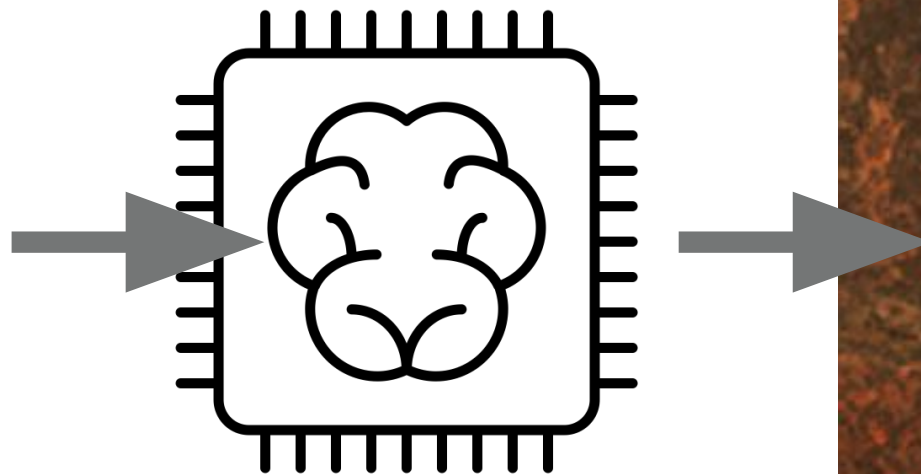


MAKING PREDICTIONS

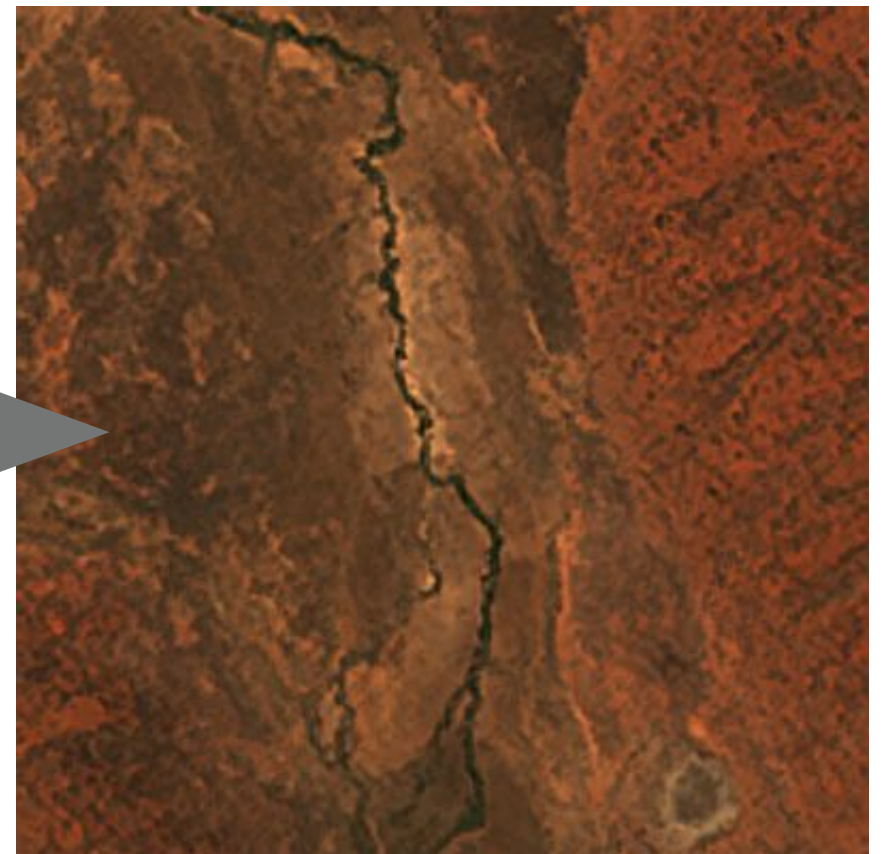
Input



*Landsat 8
25m/pixel*



Prediction



*Landsat 8
10m/pixel*

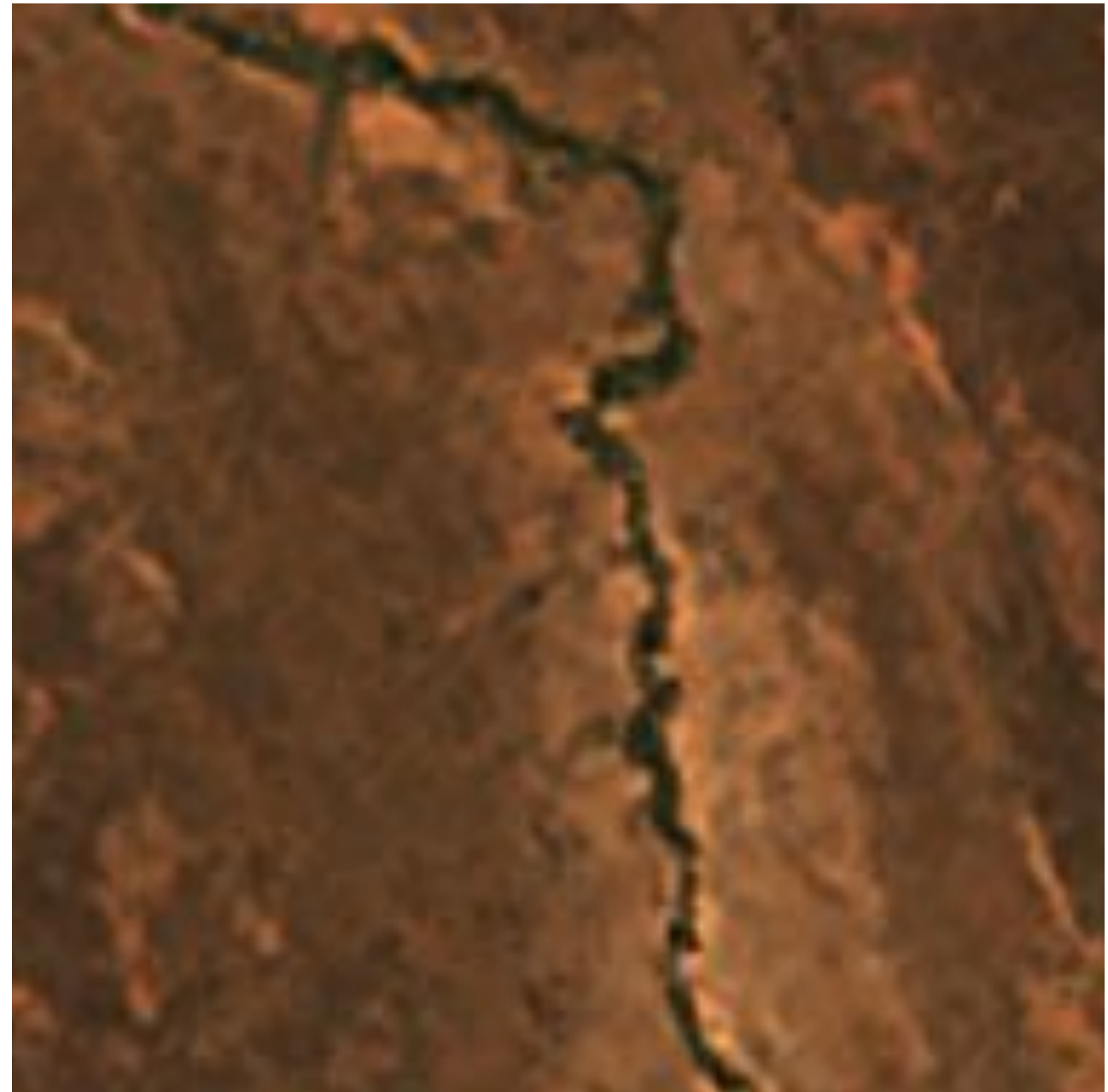
PREDICTIONS

Input



*Landsat 8
25m/pixel*

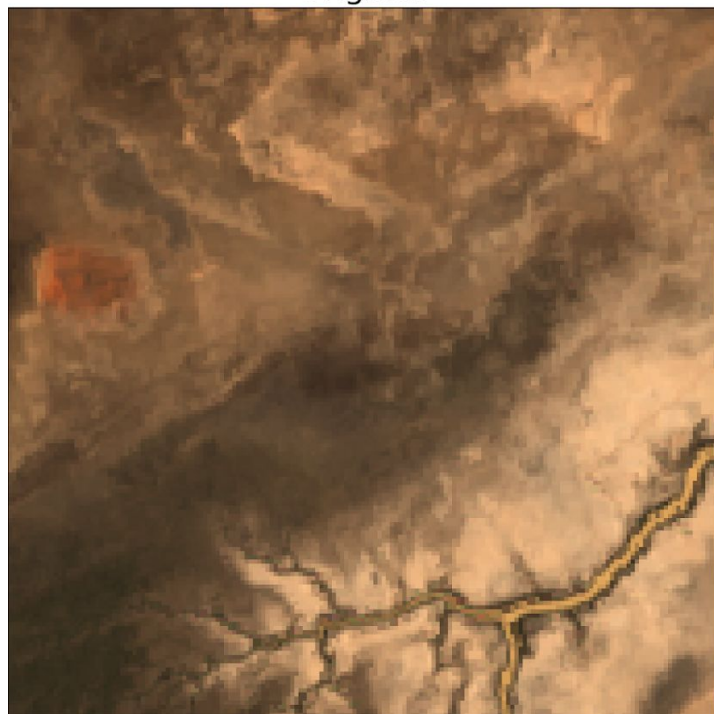
Prediction



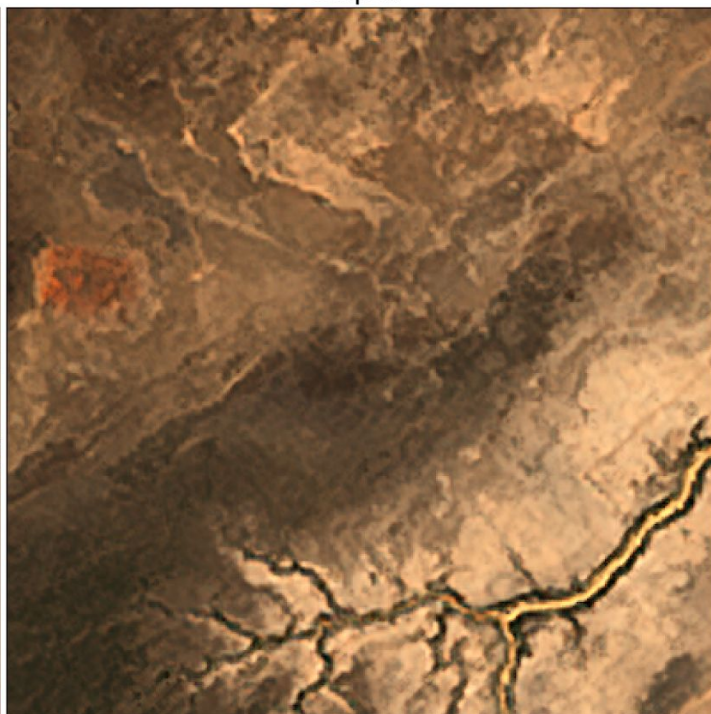
*Landsat 8
10m/pixel*

COMPARISON

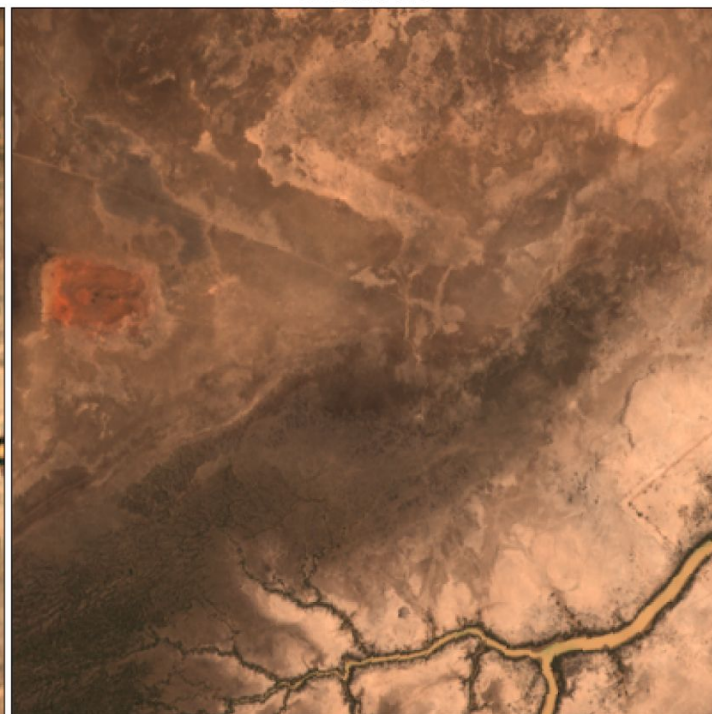
original



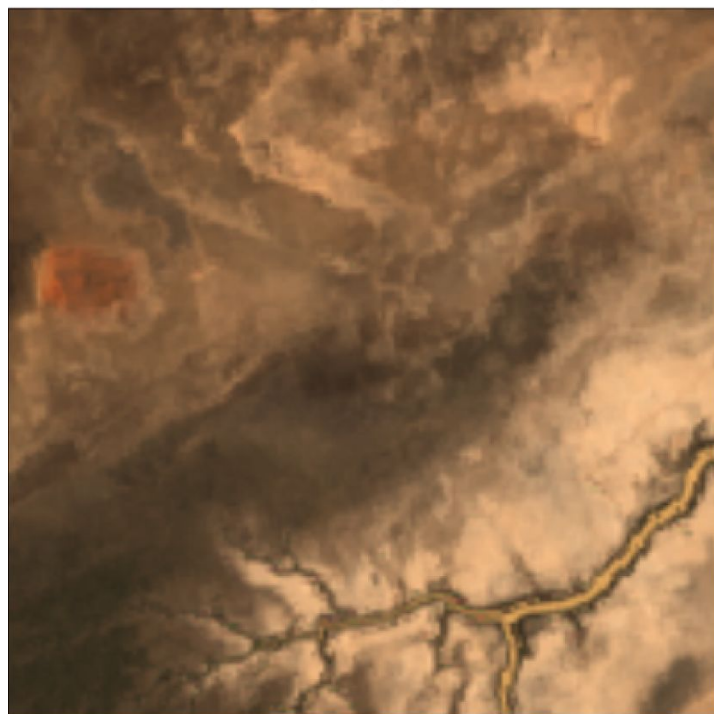
super



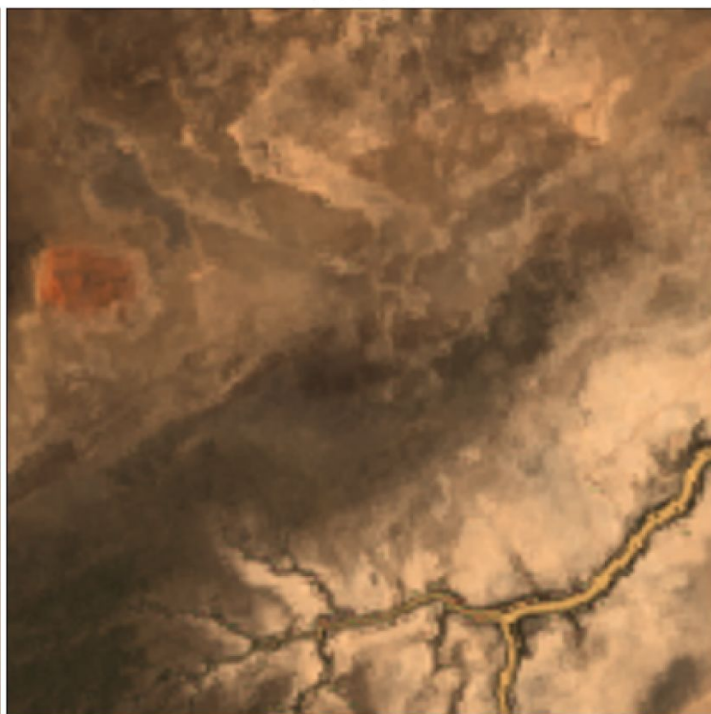
sentinel



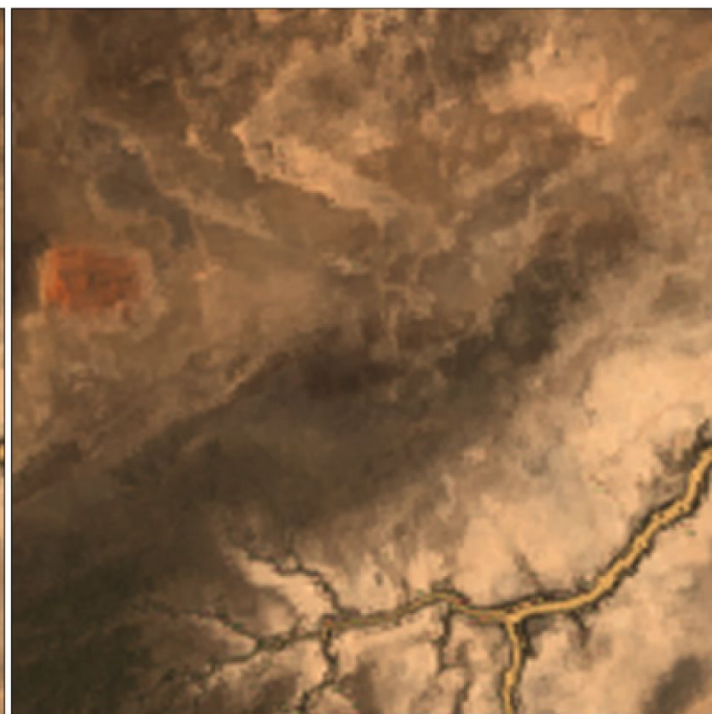
bilinear



bicubic

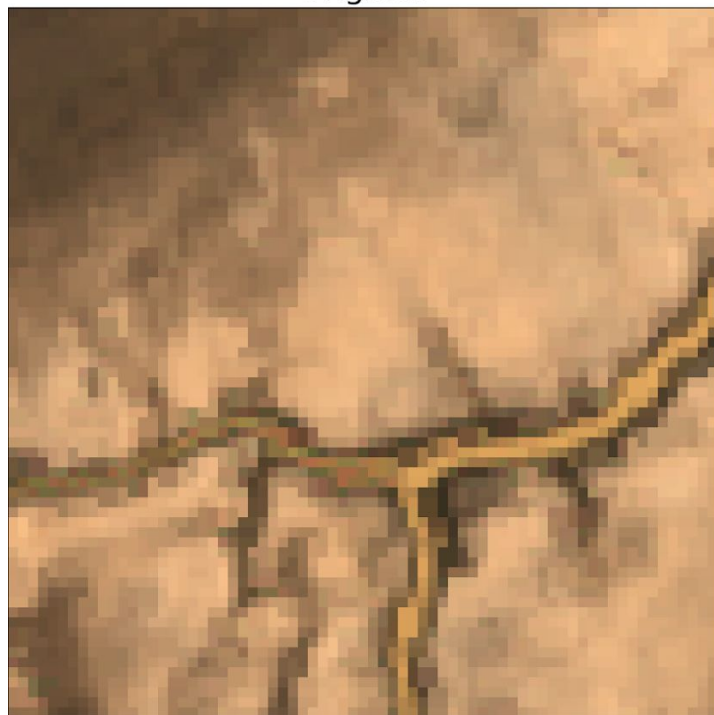


antialias

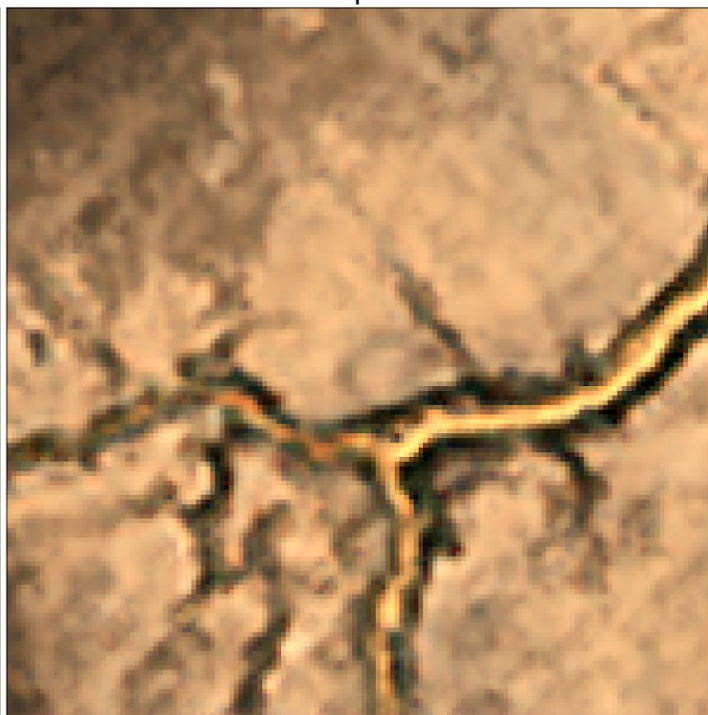


COMPARISON

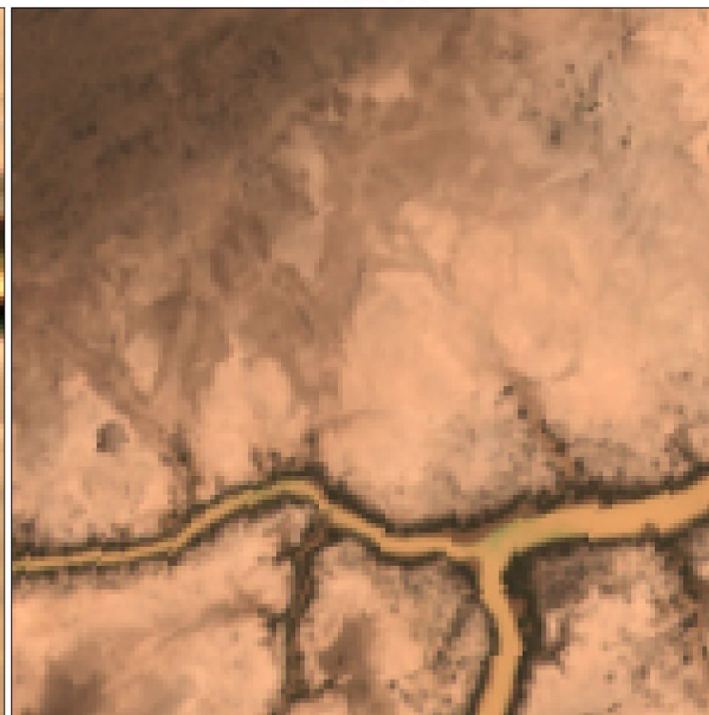
original



super



sentinel



bilinear



bicubic



antialias



WORKS ON SENTINEL!?

*Input sentinel 10m
pixels*

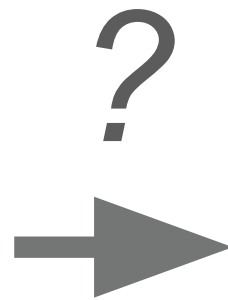
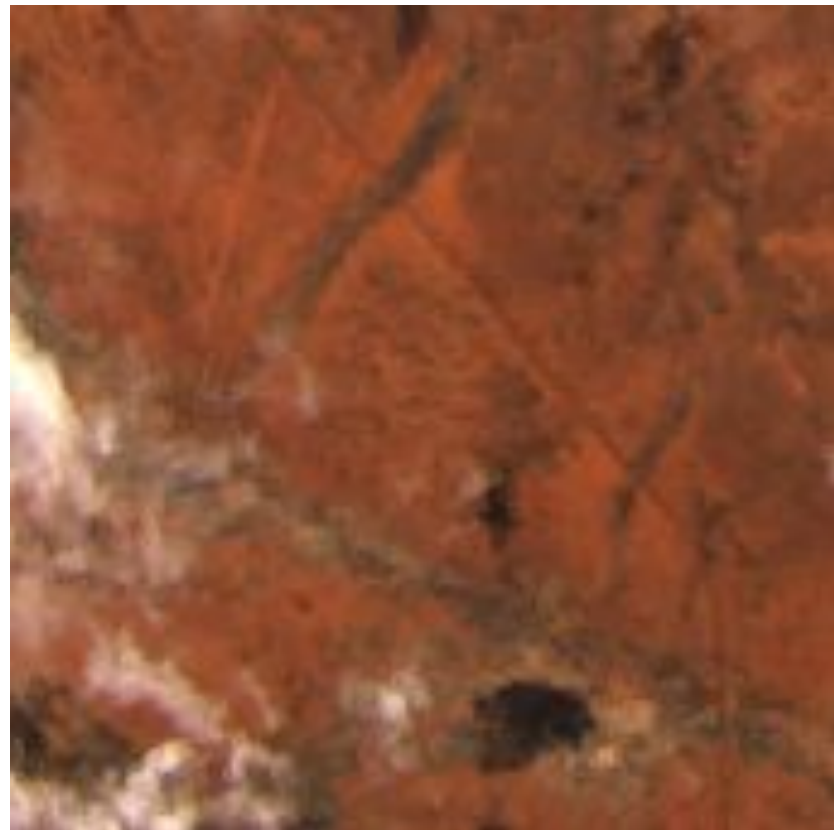
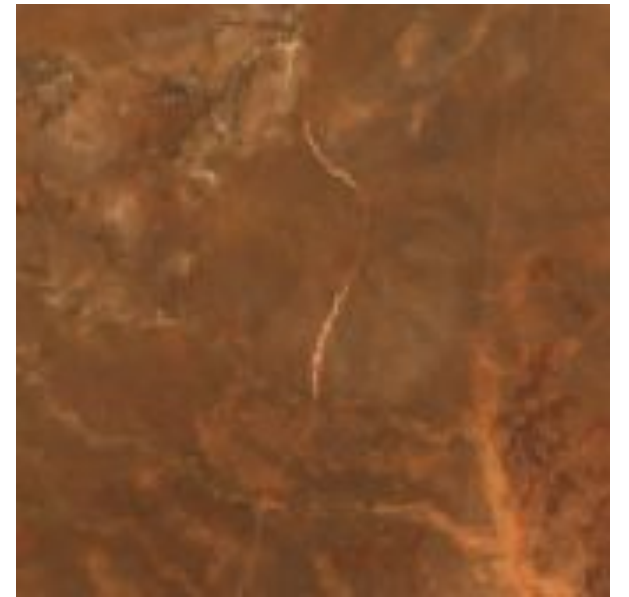


*Prediction 4m
pixels*



WHAT NEXT

- De-cloud images?
- More bands?
- More images 100,000?
- Train model from scratch?



THANKS AND FURTHER READING

Perceptual Losses for Real-Time Style Transfer and Super-Resolution

Justin Johnson, Alexandre Alahi, and Li Fei-Fei

Department of Computer Science, Stanford University
{jcjohns, alahi, feifeili}@cs.stanford.edu

Abstract. We consider image transformation problems, where an input image is transformed into an output image. Recent methods for such problems typically train feed-forward convolutional neural networks using a *per-pixel* loss between the output and ground-truth images. Parallel work has shown that high-quality images can be generated by defining and optimizing *perceptual* loss functions based on high-level features extracted from pretrained networks. We combine the benefits of both approaches, and propose the use of perceptual loss functions for training feed-forward networks for image transformation tasks. We show results on image style transfer, where a feed-forward network is trained to solve the optimization problem proposed by Gatys *et al.* in real-time. Compared to the optimization-based method, our network gives similar qualitative results but is three orders of magnitude faster. We also experiment with single-image super-resolution, where replacing a per-pixel loss with a perceptual loss gives visually pleasing results.

