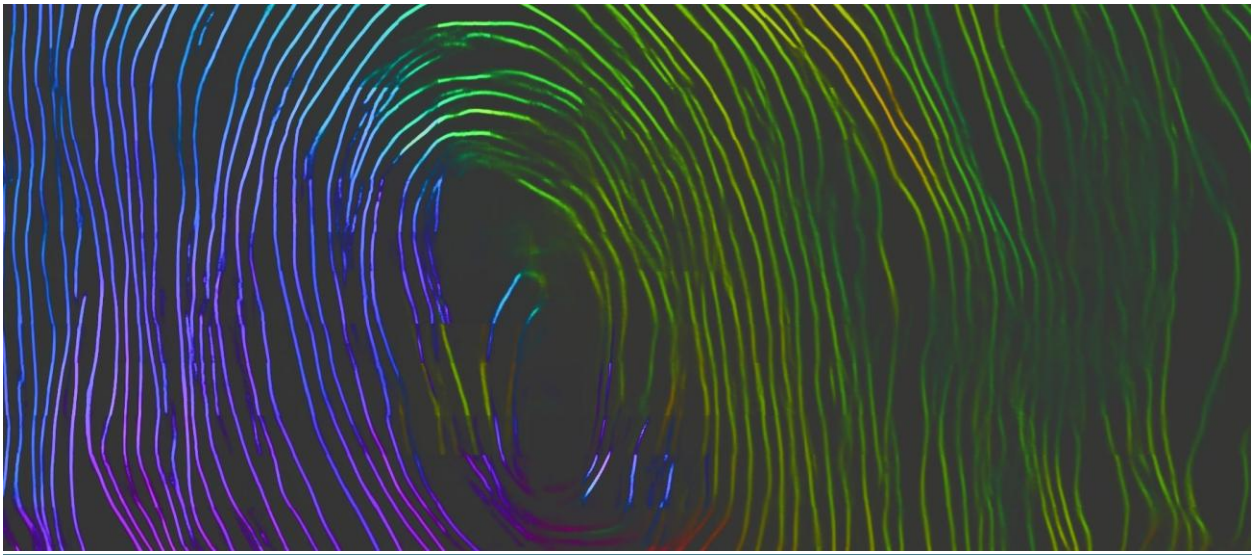
Tutorial: Ink Detection

Scanning

→



Representation

→

Segmentation and Flattening

→

Ink Detection

This tutorial gives a high-level overview on ink detection methods. Since this tutorial was written, ink detection has successfully recovered text from inside the rolled scrolls. The technical principles remain the same as what is described here.

The tutorial can be followed by the more hands-on notebooks on Kaggle and a more recent version on Colab.

Ink detection is the task of taking data from a 3D X-ray scan around a papyrus surface, and identifying the locations of the inked parts of the papyrus.

This is where one of the difficulties of the Herculaneum Papyri comes in: the ink and the papyrus have very similar densities, making it hard to detect ink in 3D X-ray scans.

- **Campfire & En-Gedi scrolls:** Dense ink shows up as brighter voxels in 3D X-ray scans, so ink detection can be done by taking the brightest pixel in some voxel region.
- **Herculaneum scrolls & fragments:** Ink is less directly visible in 3D X-ray scans, but there does seem to be data there. Machine learning models can detect it, and humans can sometimes see subtle textural patterns in the data.

In the video below Dr. Seales talks about how ink detection got started for the Herculaneum scrolls:

Not only can machine learning models detect the ink, on occasion we can see the ink directly in the 3D X-ray volumes. Here are some examples, with slices from the 3D surface volumes on the left, and infrared photos showing ink on the right (from the data paper):

Ink visible in 3D surface volumes (left: 3D volume slice; right: infrared photo), found by Stephen Parsons

You have to look closely, but the shapes are visible!

Discoveries from the community have found widespread examples of visible ink like this inside the intact scrolls. In particular the "crackle pattern" discovered by Casey Handmer has proven useful, and inspired a number of labeling approaches that successfully produce models capable of detecting ink in the scrolls.

For the purposes of the tutorial, we will use the fragment datasets, which contain ground truth ink labels made using infrared photography of the exposed writing on the surface.

At a high level, training on a fragment works like this:

From a fragment (a) we obtain a 3D volume (b), from which we segment a mesh (c), around which we sample a surface volume (d). We also take an infrared photo (e) of the fragment, which we align (f) with the surface volume, and then manually turn into a binary label image (g).

We train this model by picking a pixel in the binary label image, and sampling a subvolume around the same coordinates from the surface volume. We then backpropagate the known label data to update the model weights:

We can then use the model to predict what a label image would have looked like, from different input data than you have trained on.

Of course, in reality the label image on the right doesn't come out perfectly. Stephen Parsons' ink-id program is one example of an ML-based approach. It produces outputs like this (showing different training epochs in k-fold training/prediction):

A model learning to detect ink on a fragment, showing different training epochs

When running ink-id on all the public fragments, the results look like this (prediction left, infrared right):

Predicted label images from ink-id (left); infrared photos (right)

As you can see, some letters can be clearly seen, others not at all, and a lot of letters are somewhere in between. All fragments also have "hidden layers": pieces of papyrus that are fused to the backs of the fragments. Running the machine model on those reveals some previously unseen letters:

“Hidden layers” of papyrus, partially revealed by machine learning.

The Ink Detection Progress Prize on Kaggle was all about creating the best possible machine learning model for detecting ink within the fragments. Since then newer models have successfully uncovered ink in full scrolls (the First Letters Prize, and then the 2023 Grand Prize).

So how can a machine learning model detect ink? In the electron microscope images below (from the paper From invisibility to readability: Recovering the ink of Herculaneum), you can clearly see the difference between the inked and non-inked regions. We suspect that machine learning models are able to learn some of these features from the 3D X-ray scans.

Electron microscope pictures from the top (A and B) and the side (C) (source)

The main challenges for ink detection are:

- Model performance, getting more letters to be legible.
- Applying these models to the full scrolls.
- Reverse engineering the models to better understand the kind of patterns they are using to detect ink.
- Creating more ground truth data (e.g. “campfire scrolls” or synthetic data).

Now let’s create a model! This part of the tutorial is over on Kaggle as a notebook.

To run more advanced models on the scroll segments, check out the winning code from the First Letters Prize and the 2023 Grand Prize.