The Vesuvius Challenge: Virtually Unwrapping the Herculaneum Scrolls using Deep Learning

Matvei Kazharov
matvei@kth.se

Chanjuan Meng
chanjuan@kth.se

Andrei Vlad Dome
andreid@kth.se

May 23, 2023

Abstract

It is the year 79 in the Roman Empire when Mount Vesuvius erupts. The surrounding towns, like the well known Pompeii, are covered in hot volcanic ash. One of the affected towns, Herculaneum, is home to a giant library of ancient Roman and Greek texts that modern-day historians could only dream of reading. Unfortunately, these papyrus scrolls are completely carbonized due to being covered in hot ash. Although fortunately, their carbonization keeps them preserved for centuries to come. In 1750 these Herculaneum scrolls are discovered, and since then historians and scientists have been struggling to unwrap and read these fragile carbonized scrolls, with little success. Previous techniques which involved physically trying to unwrap the scrolls proved to have destructive effects, so with the advance of technology scientists turned to virtual unwrapping. This involves taking high-resolution 3D x-ray scans of the scrolls, determining where there is written ink, and reconstructing the unwrapped scroll. In 2019, the 3D scans of some of the scrolls were taken, and now the problem lies in detecting ink. Our project attempts to do this using deep learning and computer vision techniques, by creating a model that can learn the subtle features in the scans to determine where ink is located and reconstruct the text, all with only three small scans of scroll fragments as the ground truth labels.

1 Introduction

The carbonized scrolls have been scanned using a powerful x-ray and virtually unwrapped using a dedicated software. However, the ink is not visible on the scanned layers, hence the calls for machine learning solutions. It is suggested that neural networks may learn the ink features in the papyri and thus solve a task impossible to a human.

The challenge is hosted as a competition on Kaggle, with prizes currently ranging from $50,000 to $750,000. The grand award is given to those able to read the two intact scrolls, the dataset for which is 5.5 TB, so naturally outside of our current computational ability. However, the competition itself so far basically involves achieveing a state-of-the-art model to successfully decipher one of the three fragments (two are usually picked for training and the third for validation).

Our goal is to experiment with segmentation models and see what works for such a dataset.

2 Related Work

X-ray image segmentation is widely used in biomedical applications, particularly chest x-ray segmentation [7]. Due to often limited number of available training examples and computing power, various attempts have been made to create suitable architectures [1].

As for the scrolls, there have been methods developed to virtually unwrap them.

There is only one paper [5] relating directly to our challenge, published in May 2023, which explores deeper U-Net architectures for ink detection.

3 Data

The training data consists of grayscale 3D x-ray tomography scans of the three carbonized scroll fragments that were legible to the human eye under infrared light, done using a particle accelerator in order to achieve a very high resolution. A 3D scan is loaded as 65 2D slices, which can be visualized as splitting apart a papyrus paper into 65 layers. So in total there are 195 2D scan slices each of dimension 6218×8098. But the scroll fragments are not perfectly rectangular, so there is also black background that pads the shape of the fragment. For each of the three fragments there is also a 6218×8098 mask that highlights the area containing the fragment scan, as well as the label, represented as a binary segmentation highlighting where there is ink. A point worth mentioning is that while we have a 3D surface volume, the mask and the predictions are 2D, so 3D architectures would not do well on this. Therefore, most of the experiments use the layers of the volume as the channels, since the data is grayscale. We finally also perform the standard data processing techniques such as normalizing, zero-centering, and standardizing.

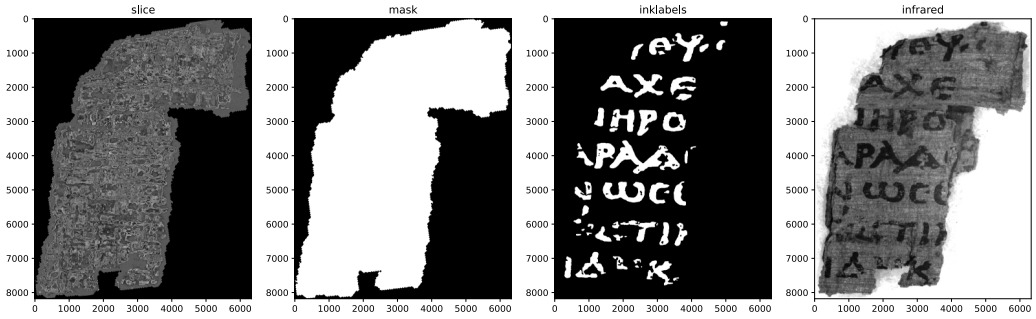

Figure 1: (From left to right) single scan slice, mask, label, infrared label

The data is passed into the neural network as 3D volumes called *patches*, that are cut-outs of the 3D scans of a chosen dimension with a chosen amount of the 65 slices. Since there is such little training data, various data augmentation techniques like rotation, shearing etc. were used in order to artificially create more training data, but this varies model to model and will be further discussed in later sections. Furthermore, the patches for the training are sampled randomly across the valid regions (the center point must be within the mask).

4 Methods

Although this problem is quite open-ended, it naturally boils down to binary image segmentation. Given 3D volumes as input data, a model outputs a predicted 2D grey-scale segmentation with showing where the ink is found.

After reading through the related papers, we decided on creating vanilla model and data pipeline from which we would experiment with hyperparameters, data augmentation techniques, and architectures. Through experimentation, we could reason about what effects certain properties had on the model's performance, metrics, training efficiency, and final prediction. Reasoning and analyzing the results of these experiments would further direct our experimentation in focusing on the variables that yielded the greatest improvements. Here we outline the variables to be experimented with.

4.1 Network Architecture & Backbones

We chose a 2D U-Net as our main architecture of choice, as current deep learning research agrees that it is an all-round solid model for image segmentation problems. It is also widely used for segmentation of x-ray images in the biomedical field, which is very similar in nature and data to our Herculaneum scroll problem. Since the input of our U-Net is 2D, we treat the input volume as a 2D image where the number of slices becomes the number of image channels.

For the encoder section of the U-Net (the down sampling convolutional layers in the first half of the U-Net), we experiment with different CNN architectures as the backbone, notably VGG-16, ResNet-18, ResNet-34, and ResNet-50. We only use the architectures and not the pre-trained weights (transfer learning) since these networks were trained on 2D images with three channels, which is incompatible with our varying channels size due to the way we process the input data.

4.2 Loss function and Metrics

One notable feature of our data is that it is very skewed, since there is relatively little ink on a fragment compared to the amount of area that is background. So using a standard metric like accuracy will not provide much useful information. There will be many correct predictions since most of the fragment consists of background, not ink, resulting in a very high accuracy even for models that very poorly predict the location of ink. Thus, we turn to a popular image segmentation metric called Intersection over Union (IoU), defined as:

$$\text{IoU} = \frac{\textit{Predicted Ink Area} \cap \textit{True Ink Area}}{\textit{Predicted Ink Area} \cup \textit{True Ink Area}} \tag{1}$$

This is much more useful since it isolates the amount of correct ink area we predicted, rather than also taking the background into consideration. So the goal is to maximize IoU, and thus we also choose a loss function that incorporates IoU, called the Binary Cross Entropy + IoU Coefficient Loss. This is a common loss function used in image segmentation, since it also benefits from the stability of BCE.

$$\text{BCE + IoU Loss} = -(y \log(p) + (1 - y) \log(1 - p)) + IoU \tag{2}$$

4.3 Optimizer

We mainly use Adam. [2] suggests SGD as a better-converging alternative, however we did not find any improvement in our application. We did limited testing in terms of learning rates, but we settled on a learning rate of 0.0007 for most of the models.

4.4 Model Hyperparameters

We experimented with different patch sizes, choosing different volume slices (`z_start`) and different amounts (`z_dim`), downscaling the image to decrease training time, as well as the usual batch related hyperparameters.

For most of the experiments, unless stated otherwise, the following hyperparameters were used:

```
patch_size = 800
downsampling = 1.0
z_dim = 40
z_start = 0
batch_size = 4
epochs = 200
steps_per_epoch = 100
```

Patches are approximately the size of one letter.

4.5 Data Augmentation and Pre-processing

Since there is little training data, augmentation is a useful technique that can also act as a kind of regularization. In our experiments, we apply random flipping, rotation and grid distortion. For this task, the 'albumentations' library was used, as it works well for image segmentation tasks.

Furthermore, we experimented with background removal and CLAHE [6]. Background removal could provide a quick and easy way to only leave the relevant parts of the image by taking an absolute difference between information-rich layers (beginning and the middle of the surface volume) and one of the last layers, which are basically uniform grey images. To increase contrast, some propose [4] convolutional architectures with CLAHE. A simple experimentation with our volumetric data may be seen on the plot below.

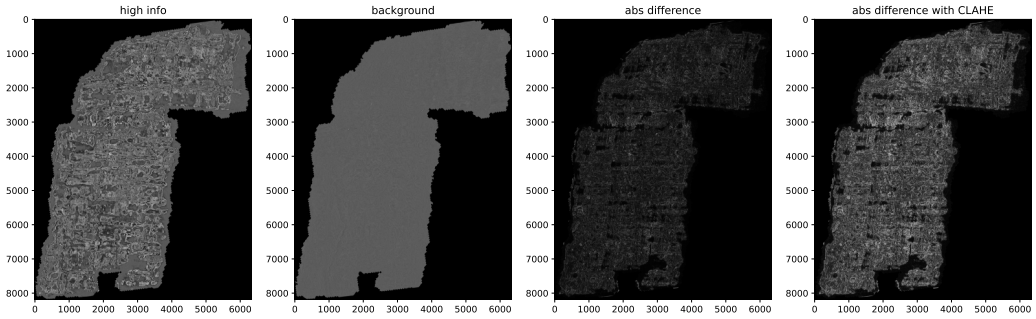

Figure 2: Background removal and contrast adjustment

4.6 Regularization

Another way we choose to avoid overfitting is by adding kernel regularization. In our baseline model we choose to avoid it, but in later experiments we use L2 Regularization as well as Batch Normalization in some of our models.

5 Experiments

We experimented with a number of different U-Net backbones, hyperparameters, and data pre-processing. All of our algorithms fail to generalize to unseen data, so the comparison is tricky. We judge the complexity of the model to be sufficient when it has the ability to overfit on the train data. Furthermore, it is meritable if the validation score correlates somewhat to the train score. The four models we present here are the models that yielded the most notable results out of the dozens we had trained.

5.1 VGG-16 U-Net, naive vanilla model

VGG-16[9] is one of the simplest backbones to the U-Net. The severe drawback is that the number of channels is fixed at 3, so we could only take 3 volumetric slices, which unsurprisingly led to poor performance. The model fails to achieve a decent IoU score even on the train set, as we can see on the plot.

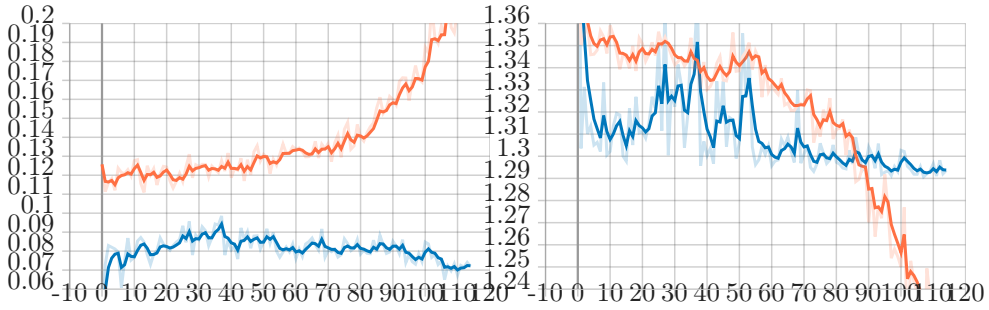

Figure 3: VGG-16 U-Net train and val IoU score (left) & loss (right) per epoch

5.2 ResNet-18 U-Net

ResNet[3] expands the possibilities introduced by VGG. This is an example of overfitting. With the ResNet-18 as the backbone, we can have a channel size of our choice. Thus, we are able to pass in a 3D volume where the number of slices chosen corresponds to the number of channels. For a simple experiment (and as suggested by the slice-by-slice analysis of the dataset, we selected 16 slices from the middle of the surface volume. The complexity of the model allows it to overfit clearly. However, it does not generalize well as seen from the way the validation IoU plateaus early on.

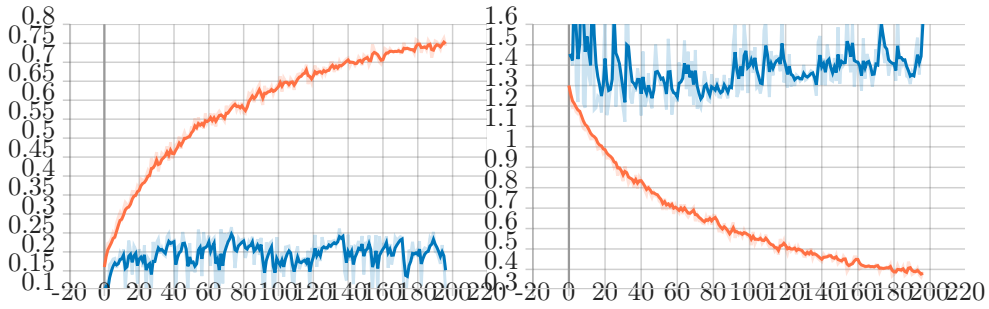

Figure 4: ResNet-18 U-Net train and val IoU score (left) & loss (right) per epoch

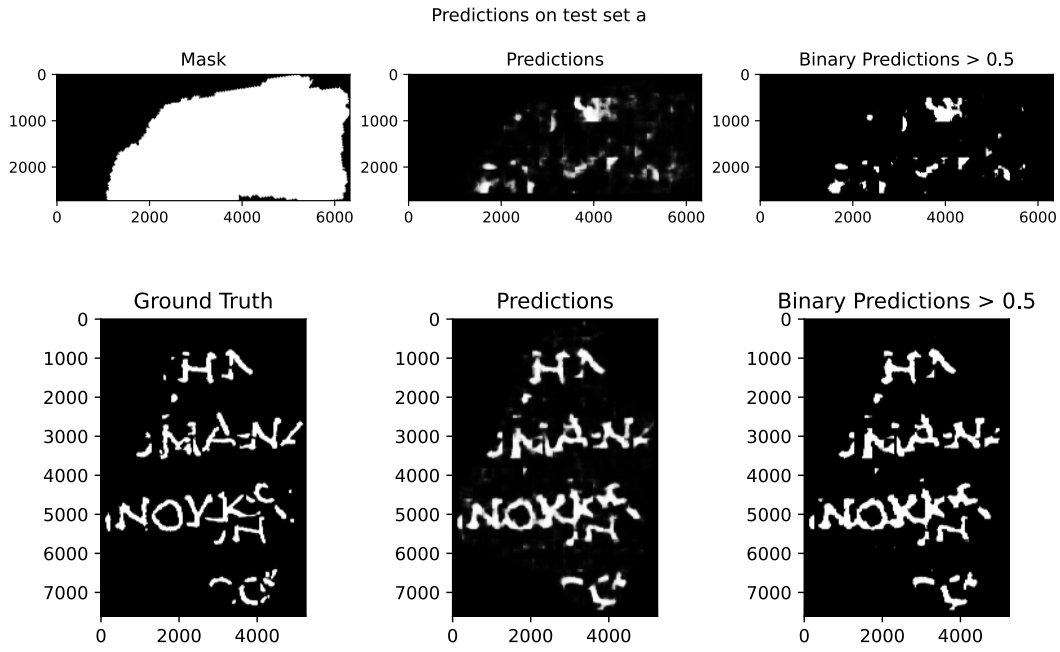

Figure 5: Predictions on test set and train set showing the model overfitting, and thus failing to generalize

5.3 ResNet-34 with 256 patch size, a good candidate

Even though we use the ResNet-34 backbone, which has more layers than the predecessor's ResNet-16 backbone, the model doesn't overfit as much. This is likely due to the fact that we use kernel and bias L2 regularization. While it still does not yield satisfactory results, it is arguably the best generalizer as seen by the correlation between the train and validation IoU. This model also achieves the highest validation IoU scores out of all the experiments.

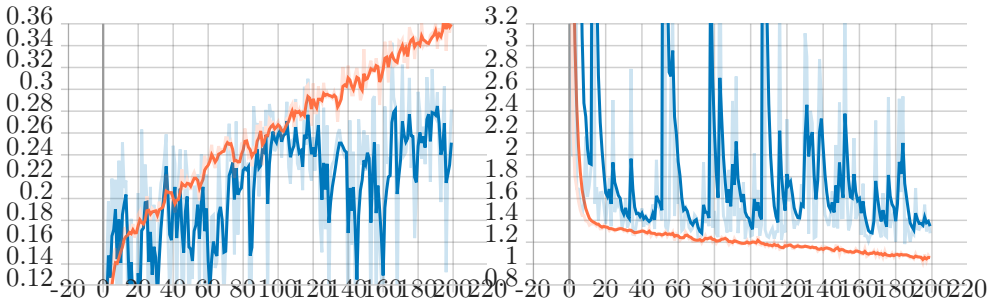

Figure 6: ResNet-34 U-Net train and val IoU score (left) and loss (right) per epoch

5.4 Attention U-Net, 2nd best

Attention-based models receive a lot of attention (pun intended) these days. One such architecture we applied is the Attention U-Net.[8] By adding attention gates on the skip connections, the U-Net model learns to focus on the informative regions of an image while disregarding irrelevant or background areas. As we aim to separate ink traces from the papyrus, this advantage of adding attention should be helpful. This U-Net has 5 levels, with filter size 16, 32, 64, 128, and 256, respectively. A patch size of 256 and layers from 23 to 38 were chosen for the input.

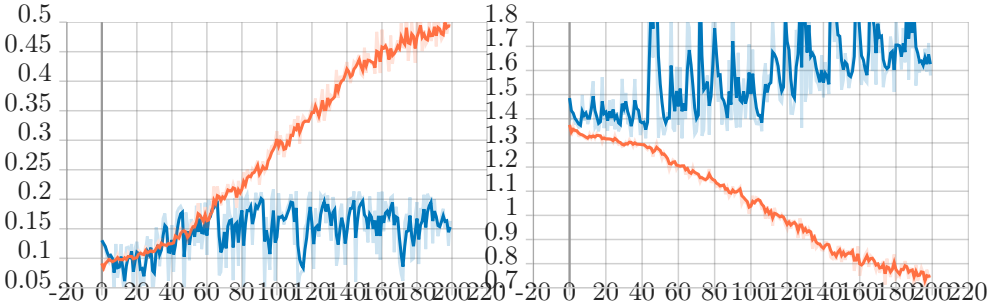

Figure 7: Attention U-Net, train and val IoU score (left) and loss (right) per epoch

As we can see, the generalization is still severely lacking, however the validation IoU score does reach 0.2.

5.5 Comparative summary

Architecture	Loss fn	Train Loss	Val Loss	Train IoU	Val IoU
VGG-16 U-Net	BCE + IoU coef	1.24	1.29	0.2	0.07
ResNet-18 U-Net	BCE + IoU coef	**0.3**	1.6	**0.75**	0.17
ResNet-34 U-Net	BCE + IoU coef	1.0	1.4	0.36	**0.24**
Attention U-Net	BCE + IoU coef	0.74	1.64	0.5	0.15

6 Conclusion

In this paper, we introduced a few approaches to the Vesuvius Ink Detection challenge. While we have not achieved satisfactory results, the work shows where improvements may be attempted. It seems that ResNet-18 and ResNet-34 based architectures are complex enough to capture the nuances of the data, but further experimentation is necessary to finalize the architecture. We also learned that changes in architecture yielded the most notable changes in performance, relative to hyperparameters which were less noticeable. If we managed to

achieve a higher baseline result with our architectures, the changes in hyperparameters would probably fine tune the results and be more noticeable. Changing the patch size also yielded differences in performance, but also was very noticeable when comparing training speeds. It is worth noting that due to random sampling imbalance in the training examples is potentially introduced with smaller patch sizes. L2 Regularization was useful, whereas Batch Normalization did not lead to improvement.

6.1 Ideas for Further Experimentation

- Using backbones trained on similar datasets, for example segmentation models for chest x-rays [2], and using these pre-trained weights in the encoder part of the U-Net (Transfer Learning).
- Training a separate network that performs a form of Optical Character Recognition on the prediction of the U-Net, "connecting the dots" of the predicted ink spots to form legible characters. This network would be trained on handwritten Ancient Greek alphabet characters. Traditionally, a human would try to interpret the models prediction and deduce the exact characters. Although a neural network could be better at finding patterns in predicted ink spots that don't seem to form a character so naturally.

6.2 Pitfalls & Improvements

One of the things that held us back was the overwhelming amount of different models, hyperparameters, and data augmentation techniques that we could experiment with as well as the constrained budget. The downside is that we were often unable to isolate a single variable in our experiments to test its effect on the performance. In the future, we may use the information we have obtained to carefully plan a set of experiments and manage obtained results in a more organized fashion.

The data we were working with was also on of the biggest challenges. It was very complex to set up, pre-process, and incorporate into a data pipeline. Our lack of background in tomography also held us back from properly understanding and analyzing the data, in order to set up our experiments and have a deeper understanding of what we were working with. If we had more time, we would definitely allocate more time to understanding our data.

References

[1] Joseph Bullock, Carolina Cuesta-Lazaro, and Arnau Quera-Bofarull. "XNet: a convolutional neural network (CNN) implementation for medical x-ray image segmentation suitable for small datasets". In: *Medical Imaging 2019: Biomedical Applications in Molecular, Structural, and Functional Imaging*. Ed. by Barjor Gimi and Andrzej Krol. SPIE, Mar. 2019. DOI: 10.1117/12.2512451. URL: https://doi.org/10.1117%5C%2F12.2512451.

[2] Gusztáv Gaál, Balázs Maga, and András Lukács. *Attention U-Net Based Adversarial Architectures for Chest X-ray Lung Segmentation*. 2020. arXiv: 2003.10304 [eess.IV].

[3] Kaiming He et al. *Deep Residual Learning for Image Recognition*. 2015. arXiv: 1512.03385 [cs.CV].

[4] Fairouz Hussein et al. *Hybrid CLAHE-CNN Deep Neural Networks for Classifying Lung Diseases from X-ray Acquisitions*. 2022. DOI: 10.3390/electronics11193075. URL: https://www.mdpi.com/2079-9292/11/19/3075.

[5] Pinxue Lin. *Deciphering Ancient Papyrus Texts: A Machine Learning Approach with Varied Architectures and Encoders*. May 2023.

[6] Akshansh Mishra. *Contrast Limited Adaptive Histogram Equalization (CLAHE) Approach for Enhancement of the Microstructures of Friction Stir Welded Joints*. 2021. arXiv: 2109.00886 [cs.CV].

[7] Nillmani et al. *Segmentation-Based Classification Deep Learning Model Embedded with Explainable AI for COVID-19 Detection in Chest X-ray Scans*. 2022. DOI: 10.3390/diagnostics12092132. URL: https://www.mdpi.com/2075-4418/12/9/2132.

[8] Ozan Oktay et al. "Attention u-net: Learning where to look for the pancreas". In: *arXiv preprint arXiv:1804.03999* (2018).

[9] Karen Simonyan and Andrew Zisserman. *Very Deep Convolutional Networks for Large-Scale Image Recognition*. 2015. arXiv: 1409.1556 [cs.CV].