Final Project of the Skoltech Deep learning Learning Course 2023

Vesuvius Challenge - Ink Detection

Bari Khairullin, Yunseok Park, Nikita Vasilev, Anastasia Gavrish, Vladislav Mityukov

Abstract

In this project, we aimed to participate in the Kaggle competition which is the sub-task of the Vesuvius challenge. We studied the methodology of the technology for Ink detection and tried several approaches to obtain a model for this specific task. As the first attempt we tried to use a pretrained model that gave us F0.5 score equal to 0.26. As you can see, some letters might be seen in the result, however they are not perfectly readable. Additionally, we explored 3D Convnet model which showed better results.

Github repo: https://github.com/vladmd22/Incas

Introduction

Vesuvius Challenge is the $1,000,000+ challenge to resurrect an ancient library from the ashes of a volcano. In this competition you are tasked with detecting ink from 3D X-ray scans and reading the contents. Thousands of scrolls were part of a library located in a Roman villa in Herculaneum, a town next to Pompeii. This villa was buried by the Vesuvius eruption nearly 2000 years ago. Due to the heat of the volcano, the scrolls were carbonized, and are now impossible to open without breaking them. These scrolls were discovered a few hundred years ago and have been waiting to be read using modern techniques.

Fig 1 - Example of Herculaneum scrolls

The main issues for this task right now are:

- **Segmenting the scrolls**. The Herculaneum scrolls are especially long, tightly wrapped, damaged, and distorted. To date, no one has successfully done a large-scale segmentation of these scrolls to identify the surfaces of all the rolled layers.

- **Finding the ink**. The ink used in the Herculaneum scrolls is radiolucent, making it difficult to see in the scans. Recently, Dr. Seales's team has trained a machine learning model which can detect the ink from subtle patterns in the 3D X-rays. This works in the fragments, but these models are not yet perfect and will probably need to be improved to work at the scale of an entire scroll.

The Kaggle competition hosts the **Ink Detection Progress Prize** ($100,000 in prizes) of this challenge, which focuses on the sub-problem of detecting ink from 3d x-ray scans of fragments of papyrus (figure 2) which became detached from some of the excavated scrolls.

Fig 2 - Fragments of scrolls for Kaggle competition

The fragments of detached papyrus were scanned at 4μm in the Diamond Light Source particle accelerator. For the submission it requires some limitations such as Notebook should run less than 9 hours run-time and also disabled internet access.
The motivation for this project is to “crack the elder scrolls” or in another way find the best model to solve this ink detection task. Recent related work has shown promising results for the original Parsons and coauthors article. Our project builds upon these developments, providing interesting results on this task.

Data organisation

The dataset contains 3d x-ray scans of four such fragments at 4μm resolution, made using a particle accelerator, as well as infrared photographs of the surface of the fragments showing visible ink. These photographs have been aligned with the X-ray scans.

Metrics

We evaluate how well your output image matches our reference image using a modified version of the Sørensen–Dice coefficient, where instead of using the F1 score, we are using the F0.5 score. The F0.5 score is given by:

$$\frac{(1+\beta^2)pr}{\beta^2 p + r} \quad \text{where} \quad p = \frac{tp}{tp+fp}, \quad r = \frac{tp}{tp+fn}, \quad \beta = 0.5$$

The F0.5 score weights precision higher than recall, which improves the ability to form coherent characters out of detected ink areas.

Algorithms and Models

We decided to focus on promising hybrid model defined below:

- Encoder is a 3D ResNet model (as described in Tran et al. 2018). The architecture has been modified to remove temporal downsampling between blocks.
- A 2D decoder is used for predicting the segmentation map.

The encoder feature maps are averaged pooled over the Z dimension before passing it to the decoder.
We also found several approaches for better solving this task:

Run-Length Encoding
Run-Length Encoding is a data compression technique that is widely used for storing and transferring data in a more efficient way (figure 3). The technique involves replacing repeated occurrences of a character or symbol with a representation of the character and a count of its occurrences. This method is useful in situations where the data contains a lot of repetitive patterns, as the technique can significantly reduce the amount of space required to store the data without losing any information.

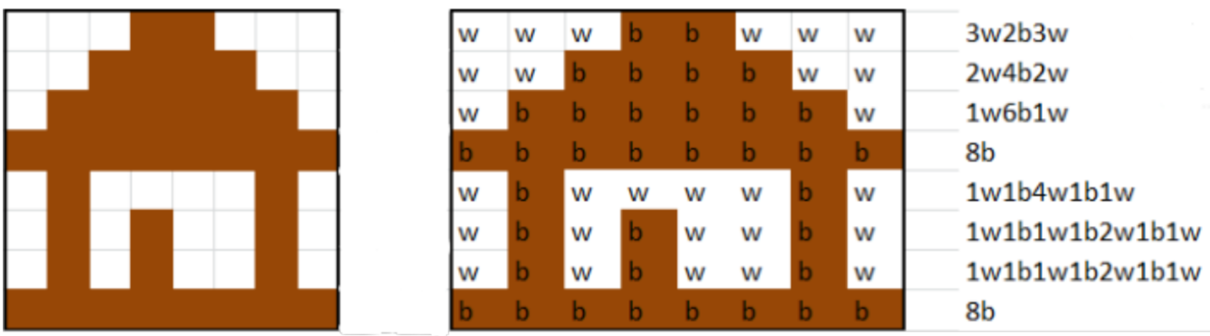


Figure 3 - Visualisation of Run-Length Encoding

L1/Hessian

L1/Hessian denoising is a popular method used for image denoising in digital image processing. The method utilizes the L1 norm, which is the sum of the absolute values of the elements in the input image, to minimize the noise present in the image and enhance its quality. In this method, the L1 norm is combined with the Hessian operator, which is a mathematical tool used to measure the curvature of an image. The denoising process begins by calculating the Hessian matrix for each pixel location in the image. The Hessian matrix is a second-order derivative matrix that is used to determine the direction and magnitude of the image curvature. The L1 norm is then applied to the Hessian matrix to identify which pixels contain the most noise. The pixels with the highest L1 norm values are then filtered out, and the remaining pixels are used to reconstruct the image. You can see denoising model below:

$$\arg\min_{\mathscr{X}} \left\{ \frac{1}{2} \left\| \mathscr{X}[h] - \mathscr{Y}[h] \right\|_2^2 + \lambda_1 \left\| \mathscr{X}[h] \right\|_1 + \lambda_2 \left\| \mathscr{H}(\mathscr{X}[h]) \right\|_2^2 \right\}$$

$$\mathscr{H}(\mathscr{X}[h]) = \begin{bmatrix} \mathscr{X}[h]_{r_1 r_1} & \mathscr{X}[h]_{r_1 r_2} \\ \mathscr{X}[h]_{r_2 r_1} & \mathscr{X}[h]_{r_2 r_2} \end{bmatrix}$$

The L1/Hessian denoising method has several advantages over other denoising techniques. One major advantage is its ability to preserve the edges in the image. Unlike some denoising methods, which tend to blur the edges and reduce the sharpness of the image, the L1/Hessian denoising method can effectively remove noise while preserving edge information, resulting in a cleaner and more visually appealing image. Another advantage of the L1/Hessian denoising method is its robustness to noise.

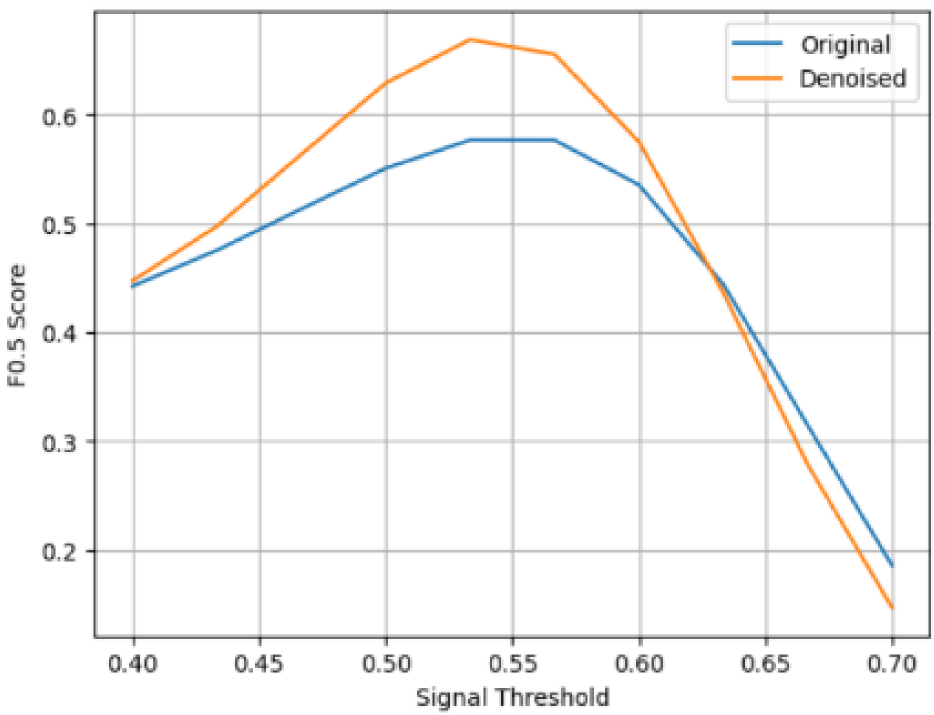


Figure 4 - Improvement of the F0.5 score after usage of Denoising model

The method can effectively remove noise from images that contain high levels of noise, making it ideal for use in noisy image environments such as medical imaging and surveillance. Although the L1/Hessian denoising method is very effective in image denoising, it does have some limitations. One major limitation is its computational complexity, which can make it time-consuming and impractical for use in real-time applications. Additionally, the method may struggle to effectively denoise images that contain textured regions or structures. Overall, the L1/Hessian denoising method is a powerful and effective method for image denoising that can produce high-quality images with sharp, well-defined edges. While it is not without its limitations, it can be a valuable tool in a range of imaging applications.

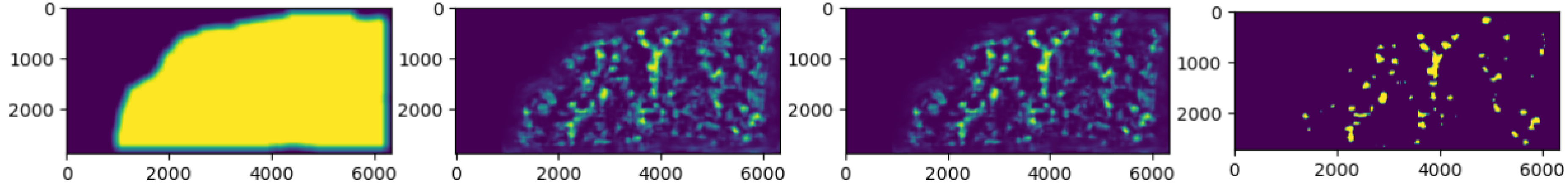
Test time augmentation

TTA is a data augmentation method used in machine learning and deep learning models to improve model accuracy and performance. TTA is applied to the test data, which means it is done during the inference phase, rather than during the training phase. The basic idea behind TTA is to create new versions of the test data by applying a variety of data augmentations such as random cropping, flipping, scaling, and rotating. These augmented samples are then fed into the trained machine learning model for prediction. The model predictions from the augmented samples are then combined, usually by averaging, to produce the final prediction. TTA can improve the accuracy and robustness of machine learning models by reducing overfitting, improving model generalization, and increasing the diversity of the test data. TTA can also help with the detection of rare objects, by ensuring that the model is tested on a wide range of object orientations and perspectives. The strength of TTA lies in the fact that it uses the same model architecture and weights trained on the original dataset, but tests the model on augmented data. TTA effectively creates more data for model testing, increasing the size of the test dataset and its diversity. Overall, TTA is a powerful technique that can improve the performance and generalization of machine learning models. It can be applied to a wide range of tasks and datasets, and is especially useful when working with limited or unbalanced datasets.

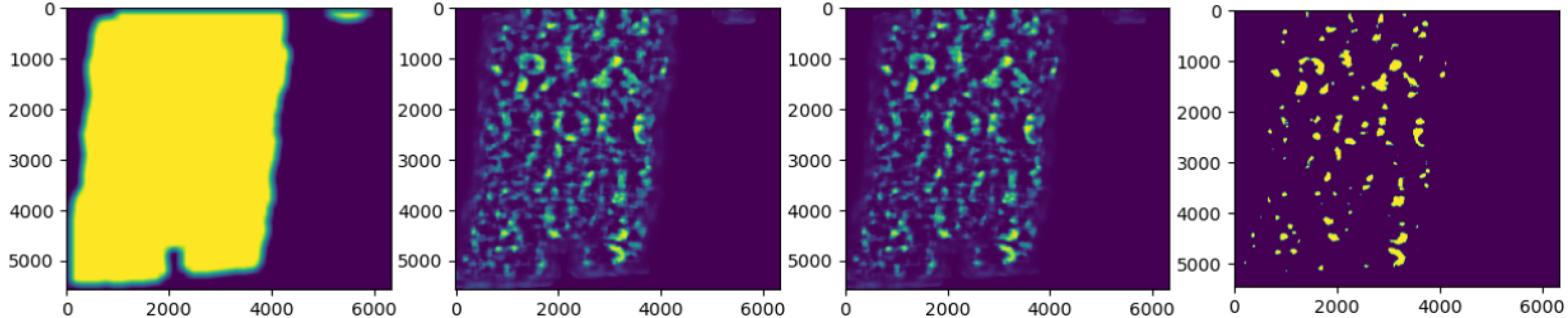
Results

On this project we obtained these results. At first we tried to use a pretrained model that gave us an F0.5 score equal to 0.26. As you can see, some letters might be seen in the result, however they are not perfectly readable (figure 5).

Mask	**Denoised image**	**Threshold 0.55**	**Predicted image**



Test dataset A
f0.5 score: **0.2130**



Test dataset B
f0.5 score: **0.2609**

Figure 5 - Results of our analysis

Also, we tried using a 3D convolutional neural network and we took the following approach: we trained the model on some subset of layers of the scan and checked the performance, so this was done to find suitable depth at which letters occur. Each time we predicted some letter and trained on everything except the letter. For the letter f0.5 score raised to 0.66 (figure 6)

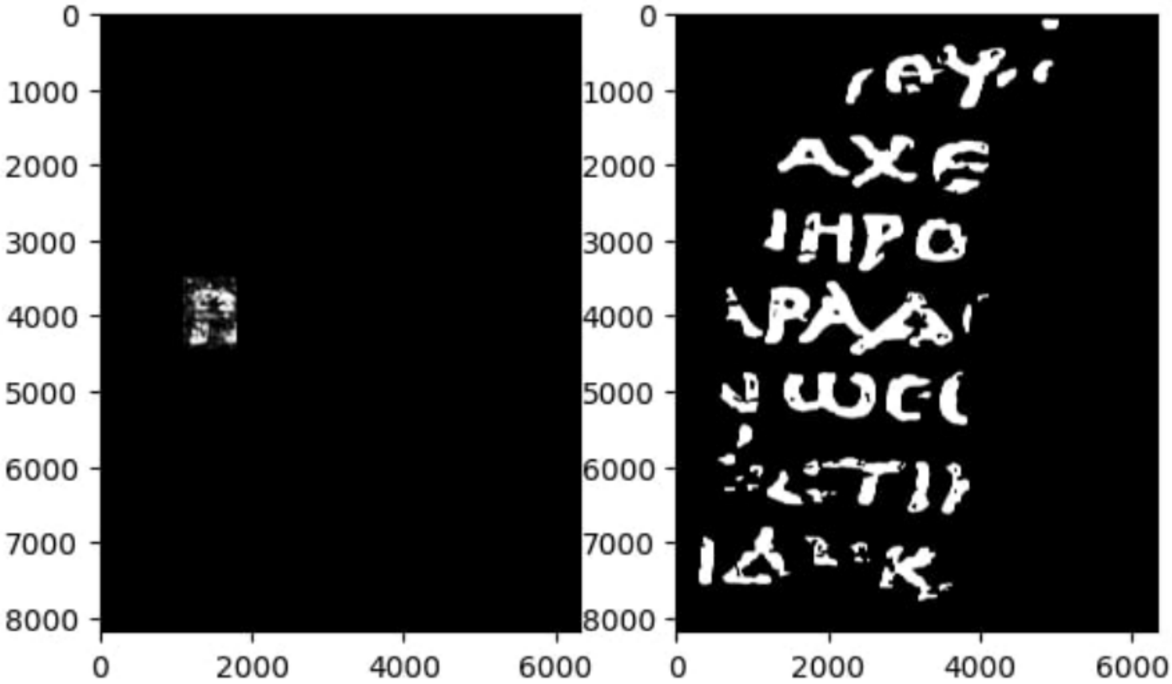


Figure 6 - Results of #D CNN model with specific layer
F0.5 score = 0.66

As you can see, the letter is much more clearly seen.
Our suggestion for further improvement as follows:
It would be great to separate the task into two steps: learn the model that somehow predicts depth of letters and the model that predicts letters themselves. Due to computational limits this task is not that easy. This approach also would perform better since we could use it on a full dataset, and not only on one fragment. Also, it is worth comparing different models such as Unet, Unet++, DeepLabV3+ and defining the optimal augmentations for our dataset.

References

1. Parsons, S., Parker, C. S., Chapman, C., Hayashida, M., & Seales, W. B. (2023). EduceLab-Scrolls: Verifiable Recovery of Text from Herculaneum Papyri using X-ray CT. *arXiv preprint arXiv:2304.02084*.
2. Ibtehaz, Nabil, and M. Sohel Rahman. "MultiResUNet: Rethinking the U-Net architecture for multimodal biomedical image segmentation." Neural Networks 121 (2020): 74-87.
3. Tran, D., Wang, H., Torresani, L., Ray, J., LeCun, Y., & Paluri, M. (2018). A closer look at spatiotemporal convolutions for action recognition. In Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (pp. 6450-6459).

Computing Infrastructure and Environment

Kaggle

Contributions of the Team Members

- **Bari Khairullin -** convnet code, repo
- **Yunseok Park -** convnet code, presentation
- **Nikita Vasilev -** both code files
- **Anastasia Gavrish -** both code files
- **Vladislav Mityukov -** repo, presentation