Dataset Collection and Baseline Results for Ink Detection on Herculaneum Papyri

Roibu Radu Gheorghe
Faculty of Automatic Control and Computers
University Politehnica of Bucharest

Silvestru Radu
Faculty of Automatic Control and Computers
University Politehnica of Bucharest

***Abstract*—We describe the dataset, preprocessing pipeline, baseline model and initial results for our ink detection project on X-ray micro-CT scans of carbonized Herculaneum papyri. We focus on a single fragment (PHercParis2Fr47), construct a patch-based training set under strict memory constraints, and train a compact 3D U-Net baseline. We report pixel-wise $F_{0.5}$, pseudo F-measure and PSNR on a held-out test set, which will serve as reference for later improvements in subsequent milestones.**

I. Introduction

The Vesuvius Challenge / EduceLab dataset contains high-resolution X-ray micro-CT scans of carbonized Herculaneum papyri, together with annotations of visible ink on exposed surfaces. Automatically detecting ink in such volumes is a challenging inverse problem due to low contrast, noise, and the thin, highly imbalanced nature of the ink structures.

In this milestone (M2), our goal is to: select and motivate a suitable dataset, implement a preprocessing pipeline, train a baseline model, and obtain initial quantitative results with appropriate metrics. These results provide a clear reference point for more advanced methods in subsequent milestones.

II. Dataset Description

We use a subset of the Vesuvius dataset corresponding to a single fragment, *PHercParis2Fr47* (Fragment 1). For this fragment, the data package includes: (i) a 3D X-ray volume of the flattened papyrus surface, stored as a stack of 2D TIFF slices (`surface_volume/*.tif`); (ii) a 2D binary ink mask (`inklabels.png`), where pixels belonging to carbon ink are labeled as 1; (iii) a 2D binary surface mask (`mask.png`), indicating the valid papyrus region and excluding air and background; and (iv) an infrared (IR) image (`ir.png`) used only for qualitative visualisation. The data originates from high-resolution micro-CT scanning and manual expert annotation of visible ink on the exposed surface.

The original CT volume has a spatial resolution of approximately 8181×6330 pixels and multiple depth slices, resulting in several gigabytes of data. This provides high-quality structural information but is impractical for direct end-to-end training under our hardware constraints (WSL with limited RAM and GPU memory). To obtain a manageable yet representative dataset, we focus on the exposed surface region by selecting four central slices from the CT stack (ink-bearing layer), apply integer subsampling by a factor of 8 along both height and width, and restrict the field of view to the area where the surface mask is active. After downsampling and cropping to the surface region-of-interest (ROI), the effective input size becomes $D = 4$ depth slices with a spatial resolution of 1636×1266 pixels.

From this ROI we construct a patch-based dataset, sampling fixed-size 3D patches and corresponding 2D ink masks. The final dataset contains 280 training patches, 60 validation patches and 60 test patches. Each patch serves as one training or evaluation example and consists of a 3D CT sub-volume (features) and the associated 2D ink mask (target). There are no missing values; potential artefacts (noise, papyrus fibres) are treated as part of the natural variability of the data.

III. Preprocessing Pipeline

The preprocessing pipeline is implemented in `src/preprocess/make_patches_frag1.py`. First, we load `inklabels.png` and `mask.png` from `data/raw/fragment1/`. If the images are stored in RGB or RGBA format, we convert them to grayscale by taking the first channel and then binarise them, mapping all non-zero values to 1 and zeros to 0. To reduce memory usage, we apply integer subsampling with a factor of 8 in both spatial dimensions, obtaining downsampled ink and surface masks.

We then compute the smallest bounding box that covers all pixels where the downsampled surface mask equals 1 and expand it with a small margin. The downsampled ink and mask images are cropped to this bounding box, yielding a compact ROI that contains only papyrus surface and ink, without large empty regions. From the CT stack in `surface_volume/`, we select four central slices corresponding to the exposed surface, downsample them with the same factor and crop them to the same ROI. The resulting volume has shape $(D, H_{\text{ROI}}, W_{\text{ROI}})$ and is normalised by clipping intensities to the $[1, 99]$ percentiles and rescaling to $[0, 1]$.

To build a balanced patch dataset, we sample patch centres from two sets of pixels within the ROI: (i) positive centres, where the ink mask equals 1, and (ii) background centres, where the surface mask equals 1 and the ink mask equals 0. Using a fixed random seed, we draw 100 positive and 100 background centres. Around each centre (y, x) we extract a 3D CT patch of size (D, P, P) and a 2D ink patch of size (P, P), with $P = 128$ pixels (reduced automatically if the ROI is smaller). All patches are shuffled and split into train/validation/test with a 70/15/15 ratio. Each patch is saved

as a compressed `.npz` file containing three arrays: `volume` (3D CT patch), `mask` (2D ink patch) and `tag` (1 for positive, 0 for background).

IV. Baseline Model

As a baseline method we employ a compact 3D U-Net architecture implemented in `src/models/unet3d_baseline.py`. The model operates on 3D patches of shape $(1, D, H, W)$, where $D = 4$ and $H = W = 128$, and outputs a logit volume of the same shape. The architecture follows the standard encoder–decoder design with skip connections: three encoder stages consisting of `DoubleConv3d` blocks (two $3 \times 3 \times 3$ convolutions with batch normalisation and ReLU) followed by 3D max-pooling, a bottleneck block with increased channel capacity, and three decoder stages with 3D transposed convolutions for upsampling and concatenation with the corresponding encoder feature maps, followed by a final $1 \times 1 \times 1$ convolution to produce a single-channel output.

Because the depth dimension is very small ($D = 4$), we perform pooling and upsampling only in the spatial dimensions (height and width) and keep the depth size constant across all layers. This avoids collapsing the depth dimension while still exploiting local 3D neighbourhoods around each pixel. During training and evaluation, we use only the central depth slice of the output volume as the 2D ink prediction, which is consistent with the data, where ink is primarily visible on the exposed surface layer.

V. Evaluation Metrics and Experimental Setup

The evaluation metrics are implemented in `src/utils/metrics.py` and are computed pixel-wise on the central slice of each patch. We use three metrics: (i) the $F_{0.5}$ score, computed between the ground-truth mask and the binarised prediction, with $\beta = 0.5$ to emphasise precision and penalise false positive ink detections; (ii) an approximate pseudo F-measure (pFM), obtained as the F_1 score between the ground-truth mask and a thresholded probability map, providing a secondary segmentation-oriented measure; and (iii) the Peak Signal-to-Noise Ratio (PSNR) between the predicted probability map and the ground-truth mask (scaled to $[0, 1]$), as a smooth similarity measure.

We train the baseline model using PyTorch for 10 epochs with the AdamW optimiser and a learning rate of 10^{-3}. The loss function is the sum of binary cross-entropy with logits and a soft Dice loss, both computed on the central slice of the output. We use a batch size of 4 (chosen to fit in GPU memory) and shuffle the training patches. The best model checkpoint is selected based on the validation $F_{0.5}$ score and saved to disk for later evaluation.

VI. Results and Analysis

Table I reports the performance of the baseline 3D U-Net on the Fragment 1 test set. The best validation score achieved during training is $F_{0.5} = 0.6608$.

TABLE I
Baseline performance on Fragment 1 (test split).

Model	$F_{0.5}$	pFM	PSNR
3D U-Net (baseline)	0.6565	0.5830	11.56

The test $F_{0.5}$ score of 0.6565 indicates that the model is able to detect a significant portion of ink pixels while maintaining a reasonably low number of false positives. The pseudo F-measure of 0.5830 confirms that the predicted segmentation masks capture a substantial fraction of the ink structures, and the PSNR value of 11.56 dB shows that the predicted probability maps are moderately close to the ground-truth masks in a continuous sense.

Qualitatively, the model performs well on thicker and higher-contrast ink strokes but struggles with very thin or low-contrast ink, which is often missed or partially fragmented. False positives tend to occur near papyrus fibres or structural artefacts with intensities similar to ink. These observations highlight the main limitations of the baseline and motivate several directions for improvement in future milestones, such as deeper or attention-based architectures, stronger data augmentation, or loss functions tailored to highly imbalanced, thin-structure segmentation. Overall, the baseline provides a reasonable starting point and a clear reference for evaluating more advanced methods in the next stages of the project.