

Fusing Forces: Deep-Human-Guided Refinement of Segmentation Masks^{*}

Rafael Sterzinger[✉], Christian Stippel[✉], and Robert Sablatnig[✉]

Computer Vision Lab, TU Wien, Vienna, AUT
{firstname.lastname}@tuwien.ac.at

Abstract. Etruscan mirrors constitute a significant category in Etruscan art, characterized by elaborate figurative illustrations featured on their backside. A laborious and costly aspect of their analysis and documentation is the task of manually tracing these illustrations. In previous work, a methodology has been proposed to automate this process, involving photometric-stereo scanning in combination with deep neural networks. While achieving quantitative performance akin to an expert annotator, some results still lack qualitative precision and, thus, require annotators for inspection and potential correction, maintaining resource intensity. In response, we propose a deep neural network trained to interactively refine existing annotations based on human guidance. Our human-in-the-loop approach streamlines annotation, achieving equal quality with up to 75% less manual input required. Moreover, during the refinement process, the relative improvement of our methodology over pure manual labeling reaches peak values of up to 26%, attaining drastically better quality quicker. By being tailored to the complex task of segmenting intricate lines, specifically distinguishing it from previous methods, our approach offers drastic improvements in efficacy, transferable to a broad spectrum of applications beyond Etruscan mirrors.

Keywords: Binarization · Interactive Segmentation · Human-in-the-Loop · Etruscan Art · Cultural Heritage

1 Introduction

With more than 3,000 identified specimens, Etruscan hand mirrors represent one of the biggest categories within Etruscan art. On the front, these ancient artworks feature a highly polished surface, whereas, on the back, they typically depict engraved and/or chased figurative illustrations of Greek mythology [5]. A primary component of their examination involves the labor- and cost-intensive task of manually tracing the artworks; an exemplary mirror is illustrated in Fig. 2 together with the sought-after tracing.

^{*} This project has received funding from the Austrian Science Fund/Österreichischer Wissenschaftsfonds (FWF) under grant agreement No. P 33721.

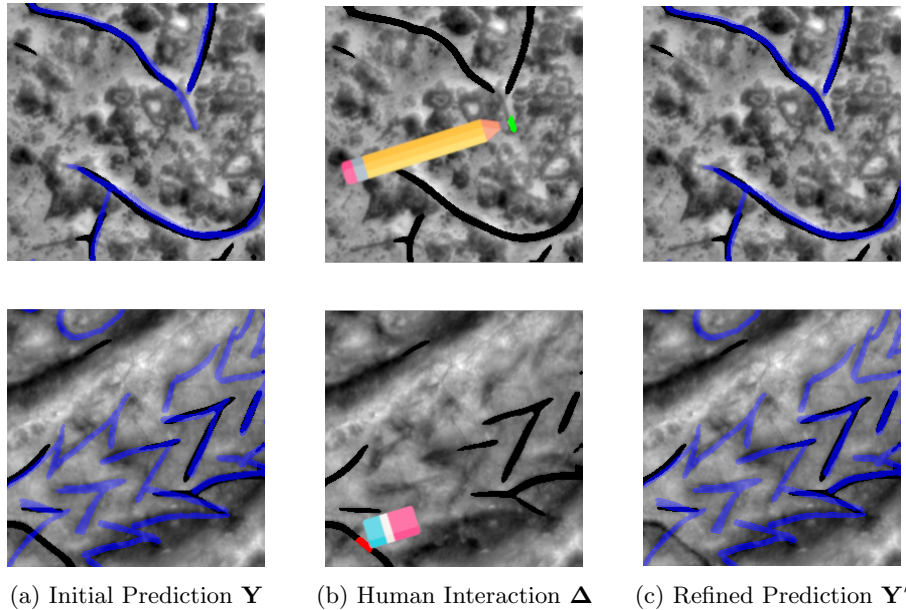


Fig. 1: Illustrating interactive refinement of segmentation masks: Starting from an initial segmentation \mathbf{Y} , the user can add (Δ^+) or erase (Δ^-) parts to bring it closer to the ground truth \mathbf{Y}^* (in blue), creating an updated mask \mathbf{Y}^Δ . Next, using a separate model conditioned on the human input Δ and \mathbf{Y} , we aim that for the refined segmentation \mathbf{Y}' it holds that $\|\mathbf{Y}' - \mathbf{Y}^*\|_1 < \|\mathbf{Y}^\Delta - \mathbf{Y}^*\|_1$.

In previous works, Sterzinger et al. [13] propose a methodology to automate the segmentation process through photometric-stereo scanning in combination with deep learning; expediting the process of manual tracing and contributing to increased objectivity. Although their segmentation model – trained on depth maps of Etruscan mirrors to recognize intentional lines over scratches – already quantitatively achieves performance on par with an expert annotator, in some instances it lags behind. Based on this, manual inspection and potential refinement by humans are still required, therefore, although alleviated, the tracing remains resource-intensive.

In this paper, we continue their line of work and propose a methodology to simplify the remaining required refinement by adding interactivity to the process: Starting from an initial prediction, we aim to reach qualitatively satisfying results as quickly as possible while keeping necessary labor to a minimum. We achieve this by training a deep neural network to refine the initial segmentation based on a series of hints, i.e., parts being added or erased, illustrated in Fig. 1.

In summary, our contribution entails the development of an interactive refinement network for improved annotation results obtained in less time, requiring less labor. Compared to refining the initial segmentation manually, fusing forces and performing the refinement interactively offers not only a drastic reduction in

labor (up to -75%) but also expedites the process by attaining significant relative performance improvements over manual labeling (up to +26%). We differentiate ourselves from prior work by proposing a methodology tailored specifically to the task of segmenting intricate lines scattered across, in our case, Etruscan mirrors versus, e.g., segmenting locally-concentrated hepatic lesions [1]. Additionally, instead of starting from scratch, we start from an initial prediction, a step required due to the non-locality of lines as otherwise labor would be drastically higher.

Finally, we provide public access to both the code and data utilized in this work (see github.com/RafaelSterzinger/etmira-interaction) to promote transparency and reproducibility.

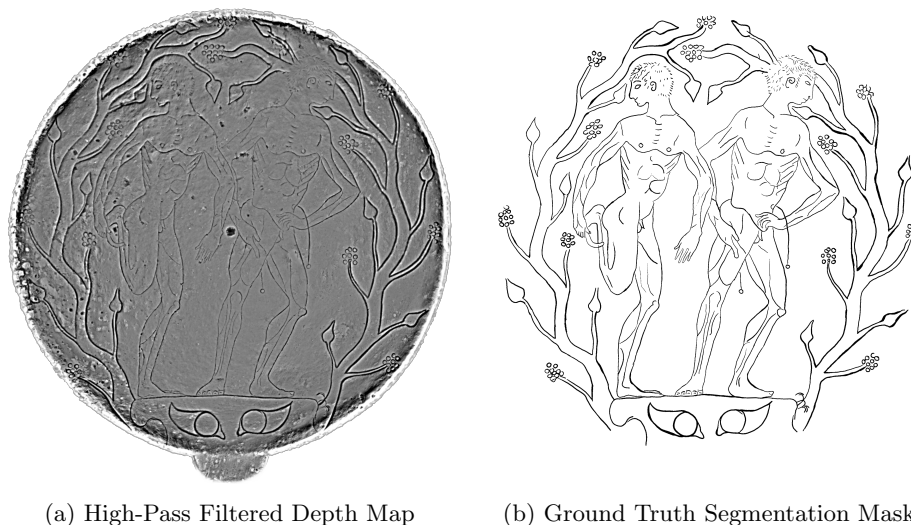


Fig. 2: Etruscan mirrors typically feature scenes from Greek mythology. During their examination, archaeologists seek to extract the drawings for visualization.

2 Related Work

Segmentation: In the field of image segmentation, techniques are led by advanced deep learning architectures such as the UNet [12], DeepLabV3++ [2], Pyramid Attention Network [7], etc. These advancements are particularly propelled by industries where precise segmentation is paramount: For example, in medical imaging, intricate segmentations are crucial for identifying vascular structures within the retina, a crucial aspect for diagnosing retinal diseases [8].

Photometric Stereo: When considering historical artifacts where the content of interest is engraved or chased into the object, as is the case with Etruscan mir-

rors, instead of RGB, modalities that capture surface details are potentially better suited. Photometric Stereo (PS), a technique introduced by Woodham [17], allows for capturing such details, providing insights into the surface geometry of an object. For instance, McGunnigle and Chantler [10] extract handwriting on paper based on depth profiles. In addition to this, PS is also employed, e.g., to detect cracks in steel surfaces [6], extract leaf venation [20], or detect air voids in concrete [16]. In the context of Etruscan mirrors, Sterzinger et al. [13] resort to a deep-learning-based segmentation approach due to the damage these mirrors have sustained. By integrating PS-scanning with deep segmentation, they learn to recognize intentional lines over scratches.

Interactive Segmentation: Independent of the segmentation methodology employed, resulting masks might not meet performance requirements and, therefore, require correction. With this regard, Li et al. [8], introduce IterNet, a UNet-based iterative approach to enforce connectivity of retinal vessels post segmentation, requiring no-external input. Similarly, interactive methods exist that incorporate human expertise within the process. Xu et al. [18] and Mahadevan et al. [9] focus on object segmentation based on mouse clicks. On the other hand and closest to our work, Amrehn et al. [1], propose an approach that refines the segmentation based on pictorial scribbles for hepatic lesions.

3 Methodology

In the following we will detail our methodology comprised of:

- the dataset; general information, splitting the data into training, validation, and testing, as well as, the preprocessing of depth maps
- the simulation of human interaction; details on the statistics of engravings, acquiring individual line segments, and the procedure for adding and erasing
- the architecture; describing the overall deep neural network used for refining the initial segmentation

3.1 Dataset

Our dataset includes a diverse array of Etruscan mirrors from public collections in Austria. It consists of PS-scans of 59 mirrors, with 53 located at the Kunsthistorischen Museum (KHM) Wien and the remaining 6 scattered throughout Austria. Annotations were acquired for 19 mirrors, encompassing 19 backsides and 10 fronts, resulting in a total of 29 annotated examples. Notably, engravings predominantly adorn the backside to avoid interference with reflectance, however, they are also occasionally found on the front, albeit with less density, near the handle or around the border. For information on the acquisition process, we refer the reader to Sterzinger et al. [13].

Dividing these annotations into training, validation, and test sets is challenging due to three factors: limited sample size, strong variations in the density

of engravings, and overall mirror conditions. Mirrors with dense engravings are prioritized for training due to the stronger learning signal they offer. To ensure fair evaluation, we select three mirrors of different conditions and engraving densities for testing: one from Wels and two from the KHM Wien. We create non-overlapping patches of size 512×512 pixels, shuffle, and split them in half to form the validation and test set of similar underlying distributions. One outlier, characterized by a different art style (points instead of lines), is excluded, leaving 25 annotated samples for training.

Preprocessing With regards to preprocessing, we employ the depth modality (which worked best according to [13]) and remove low frequencies. We accomplish this by subtracting a Gaussian-filtered version of the depth map with values capped between $\mu \pm 3\sigma$. In addition, employing the Segment Anything Model (SAM) [4], global segmentation masks are generated to identify the mirror object within a shot. We use these masks to differentiate between mirror and non-mirror parts (see Fig. 4; compare red versus green, top-left), for instance, to calculate per-channel means and standard deviations only on mirror parts which we use to normalize the input.

Addressing the lack of annotations, a per-patch inference approach is adopted. For validation and testing, non-overlapping quadratic patches measuring 512×512 pixels are extracted. Regarding our training data, we pad four pixels, since $6720 \equiv 0 \pmod{2240}$, to the original resolution ($8,964 \times 6,716$ pixels) to extract 25 overlapping tiles of size $2,988 \times 2,240$ pixels using a stride of half the size; tiles, containing no annotation, are discarded. Diversifying the dataset for each epoch, ten patches per tile are extracted, all resized to dimensions of 256×256 pixels to streamline model complexity.

3.2 Simulation of Human Interaction

In order to simulate realistic human interactions, we first look into the statistics of the annotations included in the dataset; necessary to quantitatively capture human-stroke width. Next, to refine initial predictions, we describe the process of filtering and correcting false positives and negatives. Within this, we motivate and denote the algorithm used to extract line segments. Tying all components together, we finally describe simulating human interaction: Starting from either false positives or negatives, we extract the largest error segment and provide a hint in the form of a line with width taken from the acquired statistics of the ground truth annotations.

Statistics With the goal of simulating realistic interaction, one crucial component to consider is the stroke width. For this, we look into the statistics of the annotations included in our dataset by extracting individual thickness, using Algorithm 1: Starting from a binary mask, the ground truth \mathbf{Y}^* in our case, we obtain distance information via the `euclidean_distance_transform` which, for each pixel, returns the Euclidean distance in pixels to the closest non-mask

Algorithm 1: Calculating Stroke Widths for Statistics

Data: ground truth \mathbf{Y}^*

```

def get_stroke_widths( $\mathbf{Y}^*$ ):
    distance_map  $\leftarrow$  euclidean_distance_transform( $\mathbf{Y}^*$ )
    gt_skelet  $\leftarrow$  skeletonize( $\mathbf{Y}^*$ )
    return distance_map[gt_skelet]

```

pixel. Next, employing `skeletonize` [19], we acquire a skeletonized version of the input (essential the center of lines), used to extract the thickness at each section.

From this information, we calculate initial μ and σ of the collected line widths, which we use to remove outliers (long right tail) using the two-sigma rule, keeping values within two standard deviations, to obtain final $\mu = 6.19$ and $\sigma = 1.49$. Based on this filtered set of stroke widths, we fit a Gamma distribution \mathcal{G} from which we can randomly sample realistic widths. Fig. 3 visualizes the distribution of stroke widths as well as the fitted distribution \mathcal{G} .

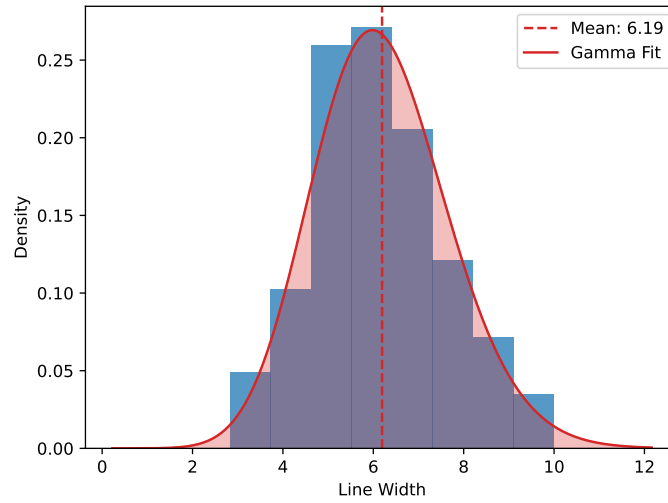


Fig. 3: Illustration of the distribution of stroke widths: After removing outliers from our data, using the two-sigma rule, we fit a Gamma distribution (shape-parameter $a = 49.13$, loc= -4.28 , scale= 0.21).

Operations In general, when an expert annotator is entrusted with the task of refining segmentation masks, one of two operations will be performed: adding missing parts or erasing superfluous ones. Simulating these operations consists of multiple steps: (1) finding an area that requires correction (we assume areas will be selected in decreasing order depending on the magnitude of correction required), (2) deciding on an operation, and (3) performing the operation. In essence, however, steps (1) and (2) go hand in hand, i.e., when deciding on an area based on error, the operation to be performed is already clear.

Let $\mathbf{Y} = f_{init}(\mathbf{X}) \in \mathbb{B}^{H,W}$ be the initial segmentation mask produced by a baseline network f_{init} based on the depth map $\mathbf{X} \in \mathbb{R}^{H,W}$. Given that we generally work with patches, using this initial mask, we find the segment that requires the most correction w.r.t. our ground truth \mathbf{Y}^* , considering the pFM (formally introduced in Section 4).

Algorithm 2: False Positive/Negative Detection for Δ^-/Δ^+

Data: ground truth \mathbf{Y}^* , prediction \mathbf{Y} , line statistics μ and σ

```

def get_add( $\mathbf{Y}^*$ ,  $\mathbf{Y}$ ):
    gt_skelet  $\leftarrow$  skeletonize( $\mathbf{Y}^*$ )
    false_negatives  $\leftarrow$  gt_skelet  $\wedge$   $\neg$   $\mathbf{Y}$ 
    return false_negatives

def get_erase( $\mathbf{Y}^*$ ,  $\mathbf{Y}$ ,  $\mu$ ,  $\sigma$ ):
    // dilate gt for more lenient detection
    expanded_gt  $\leftarrow$  dilate(skeletonize( $\mathbf{Y}^*$ ), round( $\mu + 2\sigma$ ))
    pred_skelet  $\leftarrow$  skeletonize( $\mathbf{Y}$ )
    false_positives  $\leftarrow$   $\neg$ expanded_gt  $\wedge$  pred_skelet
    return false_positives

```

Next, for the remaining steps, we propose Algorithms 2 and 3: For (2), we first employ Algorithm 2 to obtain a binary mask of missing or superfluous skeletonized segments, i.e., false positives or negatives. Note that in order to avoid the correction of minor superfluous parts, in `get_erase`, we dilate the ground truth to a constant of $\mu + 2\sigma$ s.t. only false positives which drastically diverge from \mathbf{Y}^* will be detected.

After obtaining skeletonized binary masks for false positives and negatives, for step (3), we obtain, for both, the longest line segment utilizing Algorithm 3. Within Algorithm 3, we leverage a key property of skeletonizing: In a pixel-based, skeletonized representation (i.e., one where lines have been reduced to their medial axis, which is 1-pixel wide), a single continuous line, will have exactly two neighbors in its 8-neighborhood, except for endpoints and junctions.

Let $\Delta^+ \in \{0, +1\}^{H,W}$ and $\Delta^- \in \{0, -1\}^{H,W}$ denote the missing/superfluous line segment that will be added/erased. Since these operations will be performed interactively, we summarize with Δ multiple interactions and thus contains values $\{-1, 0, +1\}$. Finally, we combine previous interactions Δ with Δ^+ or Δ^-

Algorithm 3: Obtain Edge Segments, Sorted by Length

Data: skeletonized mask \mathbf{S}

Let $K_{\text{edge}} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 10 & 1 \\ 1 & 1 & 1 \end{bmatrix}$ and $K_{\text{label}} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$.

```

def get_edges( $\mathbf{S}$ ):
    conv_skelet  $\leftarrow$  convolve( $\mathbf{S}$ , kernel= $K_{\text{edge}}$ )
    edges  $\leftarrow$  conv_skelet == 12
    // 8-connectivity
    edge_list  $\leftarrow$  label_connectivity(edges, kernel= $K_{\text{label}}$ )
    edge_list  $\leftarrow$  sort(edge_list, ord='desc')
    return edge_list

```

by leaving previously set values of ± 1 fixed, only updating 0-valued values. For simplicity, we introduce \mathbf{Y}^Δ , a quantity which denotes the union between the initial prediction \mathbf{Y} and the human interactions Δ , i.e.:

$$\mathbf{Y}_{i,j}^\Delta = \begin{cases} 1 & \text{if } \Delta_{i,j} == +1 \\ 0 & \text{if } \Delta_{i,j} == -1 \\ \mathbf{Y}_{i,j} & \text{otherwise.} \end{cases} \quad (1)$$

Interaction After introducing the three necessary steps for the adding/erasing operation, we move on to performing realistic human interactions: We continue from the previously found quantities Δ^+ or Δ^- for adding missing/erasing superfluous segments and either pick one of the two at random during training or the longer segment for maximum correction during inference. Within the skeletonized segment, we proceed by randomly sampling a sub-segment of up to eleven pixels (a parameter that we did not vary) and dilating it, based on the statistics of \mathcal{G} , with one of the following: (a) a width sampled from the distribution, (b) the mean μ , or (c) a width of $\mu - 2\sigma$.

In Section 4, options (a) and (b) will be evaluated w.r.t. validation performance, and option (c) will be used for the final evaluation on whole mirrors s.t. human interactions are with high probability aligned with \mathbf{Y}^* , i.e., reduce the risk of strokes being too wide.

Finally, using a separate network f_{iter} , trained to refine \mathbf{Y} conditioned on Δ and \mathbf{X} , we obtain a refined prediction \mathbf{Y}' . With this, we motivate the interactivity of our method: Starting over, i.e., $\mathbf{Y} \leftarrow \mathbf{Y}'$, we again find the segment that requires the most correction and update Δ with newly found Δ^+/Δ^- . A general overview of the interactivity is provided by Fig. 4, illustrating inference on a per-patch level, the initial prediction \mathbf{Y} and its refinement over time, based on Δ .

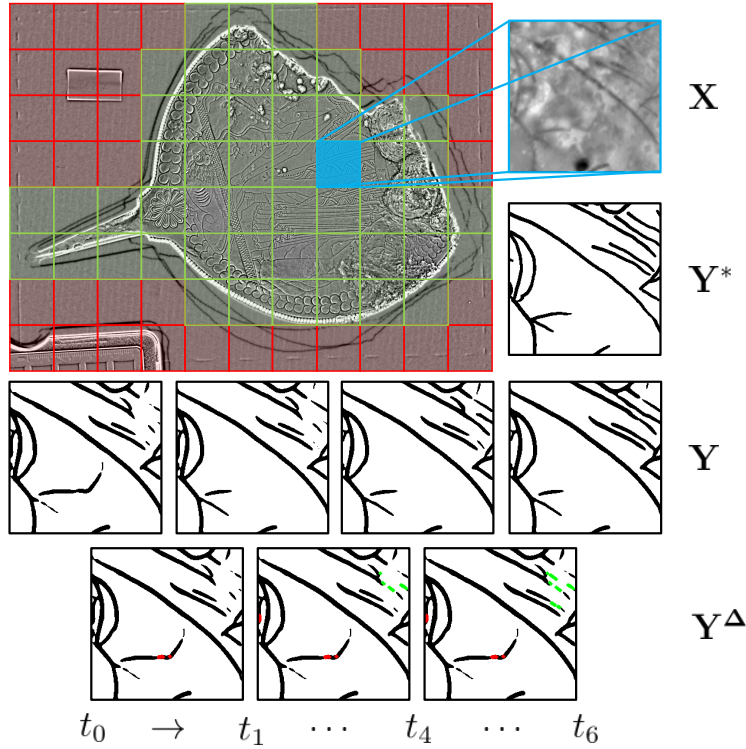


Fig. 4: An illustration of the overall methodology: In general, segmentation is performed on a per-patch level (512×512 , resized to 256×256 ; red denotes patches that are filtered a priori using SAM [4]). In an interactive paradigm, starting from the initial prediction \mathbf{Y} at timestep t_0 , based on input \mathbf{X} , a human provides hints in the form of Δ (the “union” between \mathbf{Y} and Δ is denoted with \mathbf{Y}^Δ), on which a separately trained network f_{iter} is conditioned on to produce a refined mask at timestep t_1 .

3.3 Architecture

With regards to our architecture, we employ a UNet [12] with an EfficientNet-B6 [15] following the proposal by Sterzinger et al. [13] but expand upon the input to condition the network on the (simulated) human input Δ . For clarification, the input is now comprised of a $3 \times H \times W$ tensor, including the depth map \mathbf{X} , the human input Δ , as well as the initial prediction \mathbf{Y} with all three quantities concatenated. Given that our data resources are limited, we train on a per-patch-level employing augmentations among which are rotations, flips, and shifts, optimizing the Dice loss. For the initial prediction \mathbf{Y} , we employ the exact same methodology as proposed by Sterzinger et al. [13].

4 Evaluation

In this section, we evaluate our design choices: During this process, we report the Intersection-over-Union (IoU) as well as the pseudo-F-Measure (pFM), a metric commonly used for evaluating the binarization quality of handwritten documents. It is thus well-suited for our binarization task, i.e., a task where shifting the mask by a single pixel will have a significant impact on per-pixel metrics. Compared to the standard F-Measure, the pFM relies on the pseudo-Recall (p-Recall) which is calculated based on the skeleton of \mathbf{Y}^* [11]:

$$\text{pFM}(\mathbf{Y}', \mathbf{Y}^*) = \frac{2 \times \text{p-Recall}(\mathbf{Y}', \mathbf{Y}^*) \times \text{Precision}(\mathbf{Y}', \mathbf{Y}^*)}{\text{p-Recall}(\mathbf{Y}', \mathbf{Y}^*) + \text{Precision}(\mathbf{Y}', \mathbf{Y}^*)} \quad (2)$$

Given that we work within an interactive paradigm, we are required to also provide a metric that excludes the human input Δ from the evaluation and hence report the relative pFM improvement over \mathbf{Y}^Δ , i.e.:

$$\text{pFM}_\Delta(\mathbf{Y}', \mathbf{Y}^\Delta, \mathbf{Y}^*) = \frac{\text{pFM}(\mathbf{Y}', \mathbf{Y}^*) - \text{pFM}(\mathbf{Y}^\Delta, \mathbf{Y}^*)}{\text{pFM}(\mathbf{Y}^\Delta, \mathbf{Y}^*)} \quad (3)$$

In addition, based on the fact that during training we introduce randomness, i.e., by chance, missing parts can be added (Δ^+) or superfluous ones erased (Δ^-), and that sub-segments are sampled and dilated at random, we evaluate on the test/validation set five times and report the average.

4.1 Training

Our model f_{iter} is trained on an NVIDIA RTX A5000 until convergence, i.e., no improvement $\geq 1e-3$ w.r.t. the pFM_Δ (see Equation 3) for ten consecutive epochs, using a batch size of 32 and a learning rate of $3e-4$. As a loss function, we employ a generalized Dice overlap (Dice loss) that is well suited for highly unbalanced segmentation masks [14] and optimize it using Adam [3]. Additionally, we incorporate a learning rate scheduler that also monitors the pFM_Δ on our validation set: If there is no improvement for three consecutive training epochs, the learning rate is halved.

4.2 Ablation Study

In the following, we present our ablation study, focusing on input options, different stroke widths (widths kept fixed and sampled randomly), as well as the necessity of our two operations (add and erase).

Input Options: Starting with the evaluation of different input options and their impact on the predictive patch-wise performance of f_{iter} (fixed stroke width, one interaction; results are denoted in Table 1): As expected, simply iterating over the initial prediction \mathbf{Y} (stemming from network f_{init}) results in no improvement, rendering the human an essential part of the refinement process.

Table 1: Evaluating input options and their effect on the per-patch predictive performance (fixed stroke width, one interaction): Iterating over \mathbf{Y} again does not cause improvement whereas providing Δ yields ca. +6% over \mathbf{Y}^Δ . Note that, although part of the input, we hide \mathbf{X} for clarity.

Input Modality		IoU	pFM	pFM $_\Delta$
Init. Prediction [13]	-	32.86	49.28	-
Prediction	\mathbf{Y}	32.72	49.27	-
Interaction	Δ	35.83 \pm .1	53.60 \pm .2	+5.8 \pm .23%
Both	\mathbf{Y}, Δ	36.04 \pm .2	53.44 \pm .3	+5.5 \pm .55%

Moreover, by means of human guidance, i.e., providing Δ , the network can effectively leverage additional information on missing or superfluous parts, resulting in an increase of around +6% over \mathbf{Y}^Δ . Finally, when utilizing both \mathbf{Y} and Δ , we attain a comparable improvement over \mathbf{Y}^Δ , with the difference deemed not statistically significant at a confidence level of 95%. However, the latter results in faster convergence, the reason for which we proceed with this option.

Stroke Widths: Next, we consider different options for the stroke width during the simulation of human interaction, namely: (a) keeping the stroke width constant at μ and (b) sampling it from \mathcal{G} . Our evaluation reveals that sampling does not significantly improve performance, with results showing +5.5 \pm .55% improvement for fixed width versus +4.9 \pm .35% for sampled one.

Table 2: Evaluating the impact of adding and erasing when refining mirror ANSA-1700: Employing both operations will result in the highest pFM $_\Delta$ of ca. +12%, where adding has a greater impact (ca. +8%) than erasing (ca. +2%); note that results are reported at the maximum pFM $_\Delta$.

Interaction		IoU	pFM	pFM $_\Delta$
Only Erasing	Δ^-	38.25 \pm .06	58.92 \pm .07	+1.9 \pm .12%
Only Adding	Δ^+	55.19 \pm .13	73.55 \pm .11	+8.4 \pm .16%
Both	Δ	58.41 \pm .28	76.56 \pm .16	+12.3 \pm .17%

Operations: In order to illustrate the necessity of our two operations, namely adding Δ^+ and erasing Δ^- , we perform multiple interactions until convergence and report results at the maximum attained pFM $_\Delta$. For this, we inspect an entire mirror, ANSA-1700: Employing both operations jointly yields the highest pFM $_\Delta$ of approximately +12% (redline in Fig. 5). Notably, *add* has a more significant impact (ca. +8%) compared to *erase* (ca. +2%). However, this is very

dependent on the initial prediction, thus only demonstrating that one operation supplements the other.

In summary, compared to the initial prediction \mathbf{Y} , stemming from f_{init} , providing human guidance via Δ will yield improvements exceeding \mathbf{Y}^Δ , utilizing both operations is beneficial, and augmenting stroke widths by random sampling performs worse than leaving it constant.

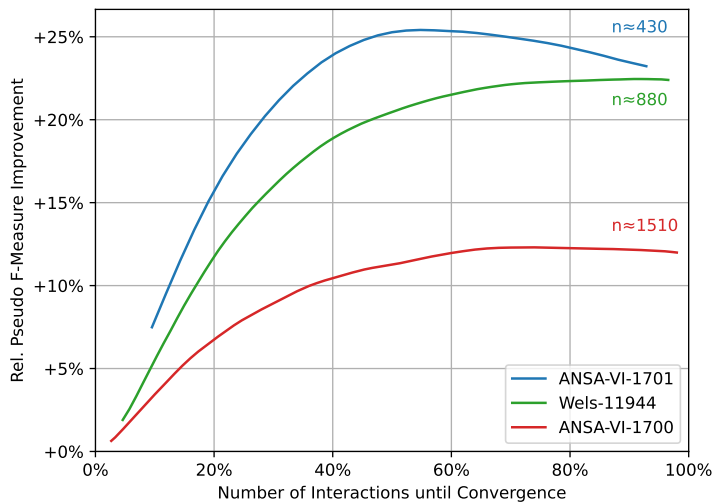


Fig. 5: An illustration of pFM_Δ , i.e., the relative pFM improvement of our method over pure manual refinement; n denotes the maximum number of human interactions within ten runs: With the relative improvement peaking at values between ca. +12% and +26%, our human-in-the-loop approach immediately overtakes manual labeling, leading to drastically better annotations earlier.

5 Results

After verifying the effectiveness of our methodology, we pick the three mirrors from our validation/test set, namely ANSA-1700, ANSA-1701, and Wels-11944, and evaluate our human-in-the-loop approach on whole mirrors, performing multiple interactions (limited to 3,000; typically requiring much less). Again, due to the introduced randomness, we repeat this process ten times and report the average result, skipping the variation as it is negligible. As described in Section 4.2, for this, we start greedily by selecting the patch with the lowest pFM, simulate adding missing/erasing superfluous parts, selecting the operation which yields a larger improvement, refine the prediction based on the additional human input,

and proceed from there until *convergence*, i.e., when neither adding nor erasing by itself increases the metric. We report the results of this in two figures: Fig. 5, which illustrates the relative pFM improvement over-performed interactions, as well as Fig. 6, which depicts potential reduction in annotation workload when employing our proposed interactive refinement paradigm.

Inspecting Fig. 5, we observe for all three mirrors significant relative improvements over the purely manual annotation baseline \mathbf{Y}^Δ when employing our method, with maximum improvements ranging from +12% to +26% depending on the mirror under consideration. Based on these results, we conclude that our human-in-the-loop approach quickly overtakes manual labeling, leading to drastically better annotations at an earlier stage. Interestingly, towards the end, relative improvement starts to decrease slightly before convergence (most notably for ANSA-1701), showcasing that at a point, the network may occasionally undo parts it had previously annotated correctly; human annotations are not impacted by this.

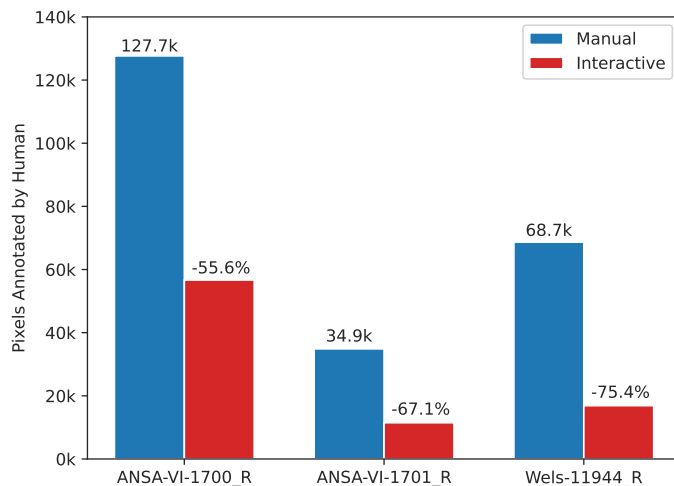


Fig. 6: An illustration of reduced workload: At convergence, our interactive approach requires drastically fewer annotated pixels to reach equal performance in pFM, resulting in a reduction of annotation effort ranging from 56% to 75%.

In Fig. 6, we directly contrast pure manual refinement against our interactive approach. For this, we determine the maximum attained pFM, calculated using \mathbf{Y}^Δ , which corresponds to the final simulated human interaction. We then compare the amount of required human input to the human input necessary to reach equal or higher performance using our proposed method. By doing this, we

are able to report a notion of workload reduction: Depending on the mirror under inspection, annotation requirements will experience a reduction ranging from around -56% to -75%, positioning our model well to be employed for simplifying the task of correcting erroneous segmentation masks.

6 Limitations and Future Work

While our proposed method shows promising results, it is important to acknowledge its limitations: At the moment human guidance aids refinement only locally, i.e., modifications happen just in the vicinity of the provided annotation. Moving forward, one could focus on further refining our methodology by exploring additional techniques to enhance efficiency. For instance, it would be meaningful to investigate the integration of quickly trainable learning algorithms, such as Gaussian processes which can immediately be adapted to newly provided annotation and thus allow for global adjustments, potentially further reducing the amount of human input required. Additionally, leveraging Gaussian processes is accompanied by the option of active learning strategies, which could allow the identification and annotation of patches where the model is most uncertain with the chance of expediting refinement further.

7 Conclusion

In summary, our research addresses the labor-intensive process of manually tracing intricate figurative illustrations found, for instance, on ancient Etruscan mirrors. In an attempt to automate this process, previous work has proposed the use of photometric-stereo scanning in conjunction with deep neural networks. By doing so, quantitative performance comparable to expert annotators has been achieved; however, in some instances, they still lack precision, necessitating correction through human labor. In response to the remaining resource intensity, we proposed a human-in-the-loop approach that streamlines the annotation process by training a deep neural network to interactively refine existing annotations based on human guidance. For this, we first developed a methodology to mimic human annotation behavior: We began by analyzing annotation statistics to capture stroke widths accurately and proceeded by introducing algorithms to select erroneous patches, identify false positives and negatives, as well as correct them by erasing superfluous or adding missing parts. Next, we verified our design choices by conducting an ablation study; its results showed that providing human guidance will yield improvements exceeding pure manual annotation, utilizing both operations is beneficial, and augmenting stroke widths by random sampling performs worse than leaving it constant. Finally, we evaluated our method by considering mirrors from our test and validation set. Here, we achieved equal quality annotations with up to 75% less manual input required. Moreover, the relative improvement over pure manual labeling reached peak values of up to 26%, highlighting the efficacy of our approach in reaching drastically better results earlier.

References

1. Amrehn, M., Gaube, S., Unberath, M., Schebesch, F., Horz, T., Strumia, M., Steidl, S., Kowarschik, M., Maier, A.K.: UI-Net: Interactive Artificial Neural Networks for Iterative Image Segmentation Based on a User Model. arXiv preprint arXiv:1709.03450 (2017)
2. Chen, L.C., Zhu, Y., Papandreou, G., Schroff, F., Adam, H.: Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation, p. 833–851. Springer International Publishing (2018)
3. Kingma, D., Ba, J.: Adam: A Method for Stochastic Optimization. In: International Conference on Learning Representations (ICLR). San Diego, CA, USA (2015)
4. Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W.Y., Dollar, P., Girshick, R.: Segment Anything. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 4015–4026 (Oct 2023)
5. Kluge, S.: Through the Looking-Glass: Der etruskische Spiegel VI 2627 aus dem Bestand des Kunsthistorischen Museums als Fallbeispiel für die rituelle Manipulation und Defunktionalisierung im Kontext der Grablege in der etruskischen Kultur. In: Weidinger, A., Leskovar, J. (eds.) Interpretierte Eisenzeiten. Fallstudien, Methoden, Theorie. Tagungsbeiträge der 10. Linzer Gespräche zur interpretativen Eisenzeitarchäologie, pp. 241–254. Studien zur Kulturgeschichte von Oberösterreich, Folge 55, Linz (2024)
6. Landstrom, A., Thurley, M.J., Jonsson, H.: Sub-Millimeter Crack Detection in Casted Steel Using Color Photometric Stereo. In: 2013 International Conference on Digital Image Computing: Techniques and Applications (DICTA). IEEE (Nov 2013)
7. Li, H., Xiong, P., An, J., Wang, L.: Pyramid Attention Network for Semantic Segmentation. In: British Machine Vision Conference 2018, BMVC 2018, Newcastle, UK, September 3-6, 2018. p. 285. BMVA Press (2018)
8. Li, L., Verma, M., Nakashima, Y., Nagahara, H., Kawasaki, R.: IterNet: Retinal Image Segmentation Utilizing Structural Redundancy in Vessel Networks. In: 2020 IEEE Winter Conference on Applications of Computer Vision (WACV). IEEE (Mar 2020)
9. Mahadevan, S., Voigtlaender, P., Leibe, B.: Iteratively trained interactive segmentation. In: British Machine Vision Conference 2018, BMVC 2018, Newcastle, UK, September 3-6, 2018. p. 212. BMVA Press (2018)
10. McGunnigle, G., Chantler, M.: Resolving handwriting from background printing using photometric stereo. *Pattern Recognition* **36**(8), 1869–1879 (Aug 2003)
11. Pratikakis, I., Gatos, B., Ntirogiannis, K.: ICFHR 2012 Competition on Handwritten Document Image Binarization (H-DIBCO 2012). In: 2012 International Conference on Frontiers in Handwriting Recognition. IEEE (Sep 2012)
12. Ronneberger, O., Fischer, P., Brox, T.: U-Net: Convolutional Networks for Biomedical Image Segmentation, p. 234–241. Springer International Publishing (2015)
13. Sterzinger, R., Brenner, S., Sablatnig, R.: Drawing the Line: Deep Segmentation for Extracting Art from Ancient Etruscan Mirrors. In: 2024 ICDAR International Conference on Document Analysis and Recognition, submitted (2024)
14. Sudre, C.H., Li, W., Vercauteren, T., Ourselin, S., Jorge Cardoso, M.: Generalised Dice Overlap as a Deep Learning Loss Function for Highly Unbalanced Segmentations, p. 240–248. Springer International Publishing (2017)

15. Tan, M., Le, Q.V.: EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. In: Chaudhuri, K., Salakhutdinov, R. (eds.) Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA. Proceedings of Machine Learning Research, vol. 97, pp. 6105–6114. PMLR (2019)
16. Tao, J., Gong, H., Wang, F., Luo, X., Qiu, X., Huang, Y.: Automated image segmentation of air voids in hardened concrete surface using photometric stereo method. *International Journal of Pavement Engineering* **23**(14), 5168–5185 (Nov 2021)
17. Woodham, R.J.: Photometric method for determining surface orientation from multiple images. *Optical Engineering* **19**(1) (Feb 1980)
18. Xu, N., Price, B.L., Cohen, S., Yang, J., Huang, T.S.: Deep Interactive Object Selection. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016. pp. 373–381. IEEE Computer Society (2016)
19. Zhang, T.Y., Suen, C.Y.: A fast parallel algorithm for thinning digital patterns. *Communications of the ACM* **27**(3), 236–239 (Mar 1984)
20. Zhang, W., Hansen, M.F., Smith, M., Smith, L., Grieve, B.: Photometric stereo for three-dimensional leaf venation extraction. *Computers in Industry* **98**, 56–67 (Jun 2018)