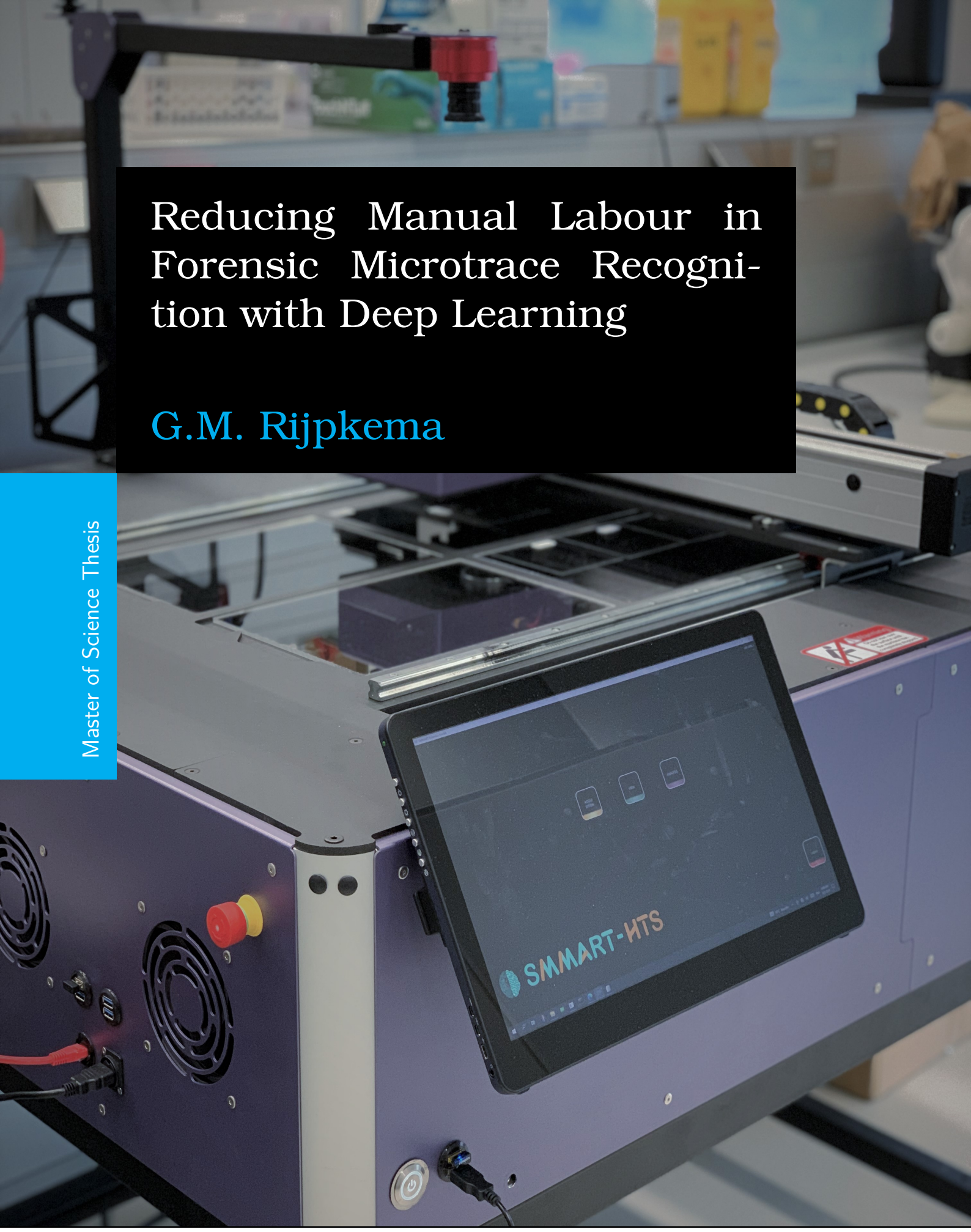
Reducing Manual Labour in Forensic Microtrace Recognition with Deep Learning

G.M. Rijpkema

Master of Science Thesis



TU Delft Delft University of Technology

Delft Center for Systems and Control

Reducing Manual Labour in Forensic Microtrace Recognition with Deep Learning

Master of Science Thesis

For the degree of Master of Science in Systems and Control at Delft University of Technology

G.M. Rijpkema

February 17, 2024

Faculty of Mechanical, Maritime and Materials Engineering (3mE) · Delft University of Technology

The work in this thesis was supported by the Netherlands Forensic Institute. Their cooperation is hereby gratefully acknowledged.

Abstract

Forensic microtrace investigation relies on a time- and labour-intensive process of manually analysing samples via microscopy. To aid forensic experts in their investigations, an image recognition model for microtrace localisation and classification is needed. This work investigates the trace recognition accuracy that can be achieved by analysing images captured with automated microscopy through deep learning. Fibres, hairs, skin, glass and sand are pixel-wise classified in microscopy scans of tape-lift samples. As deep learning requires extensive amounts of annotated training data, pretraining is investigated to minimise the required annotation workload. ImageNet pretraining, pretraining with self-supervised learning and a sequential application of these approaches are tested. It is found that pretrained models are able to reduce the required annotated data twofold compared to models trained from scratch, while retaining the prediction accuracy. While the ImageNet-pretrained models outperform the self-supervised-pretrained models, the highest accuracy is achieved by combining the two approaches. With this, traces are recognised with a mean intersection over union of 0.56 when training on only 2.2 dm^2 of annotated tape lift scans.

Table of Contents

List of Figures

List of Tables

Acknowledgements

This work would not have been possible without the endless support of my team of supervisors: Carlas Smith, Jaap van der Weerd, Dylan Kalisvaart and Serafim Korovin. The combination of your areas of expertise gave me endless inspiration throughout this project.

Carlas, your critical mindset, expertise and support were of great value in this project. Jaap, your knowledge, expertise and innovative ideas helped shape this project and its value for forensics. The new perspectives from our open dialogues guided me through the world of forensics and offered fresh perspectives on possible directions and solutions, which you allowed me the freedom to pursue. Dylan and Serafim, I want to thank you for the extensive feedback, trust and our lively discussions. Our meetings always opened my eyes and allowed me to keep on track in digging this work into rigorous research. Your feedback helped me both on an academic and personal level.

Next to my supervisors, I would like to express my appreciation to Daniel for the feedback and support with the computing cluster, to Anna with the help with the Shuttle, to my other colleagues at NFI for the welcoming and warm atmosphere and to Mark and Erwin for connecting me to the NFI, facilitating the context of this project and the fruitful collaboration.

Last but certainly not least, I want to thank my friends and family. My parents and brother for the support, my housemates for listening to the challenges I experienced during my work and motivating me, my study friends, sports friends and friends from the TU Delft Solar Boat Team, for the relaxing moments, the energising moments and of course the moments of blowing off steam. Thank you all!

Delft, University of Technology — G.M. Rijpkema
February 17, 2024

Chapter 1

Introduction

For a safe society, a healthy rule of law is crucial. Execution of justice encourages respect for the law. Furthermore, it protects the rights of individuals and ensures accountability. A key part in the rule of law is finding the truth. Institutes such as the Netherlands Forensic Institute (NFI) provide key information for justice by examining and analysing traces from crime scenes and other locations. These traces can help with the prosecution of criminal offenders and the clearing of innocent persons from suspicion by providing information on people, objects, locations and actions [59, 66, 86].

Traces can take various forms. Well-known examples include DNA traces, footprints and fingerprints. Although less well-known, traces such as textile fibres, paint particles, hairs and glass particles can bear large forensic value [66]. As shown in Figure 1-1, these traces typically have sizes in the order of $10^1 - 10^3$ µm and can barely be seen by the human eye. Compared to the size of a typical crime scene, they are a needle in a haystack.

Experts aim to find traces with multiple methods. One of the methods used to recover traces from bodies, surfaces, and objects is tape lifting. Here, transparent adhesive tapes are applied to the area of interest and then lifted off to extract the traces. Then, the recovered traces are analysed with various forms of microscopy [60]. In one-to-one taping, a large number of tapes is applied to fully cover for example the body of a murder victim. This allows mapping the locations of the traces, which can yield crucial information on the crime [18]. Due to its large analysis workload, it is only applied in few cases [18]. More often, a strict selection is made on areas and traces to analyse, tapes are re-applied to increase the number of traces per tape [51], or trace investigation is fully absent.

A step has been made to streamline the analysis of microtrace samples with the European *Shuttle* project [70]. In this project, an automated microscope was developed able of

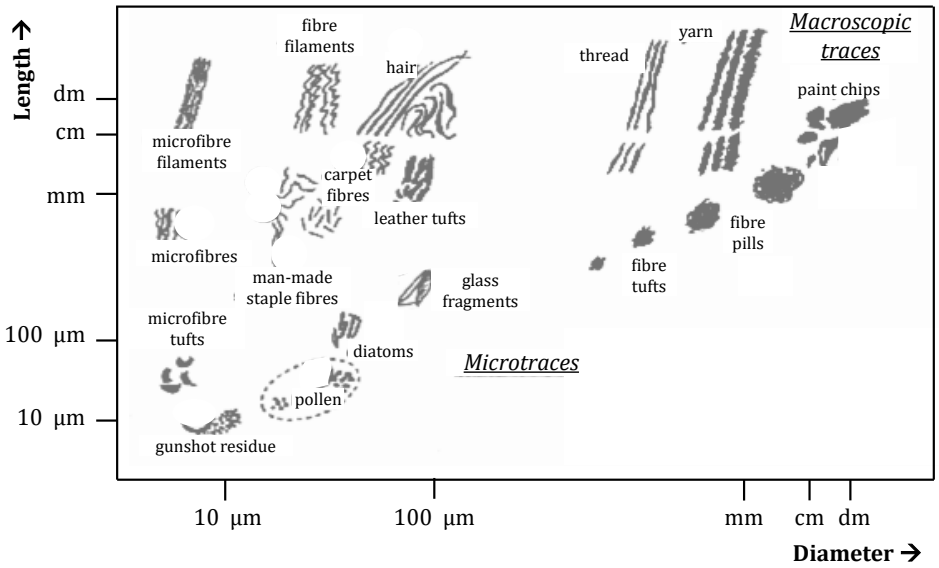


Figure 1-1: Comparison of different types of microtraces in length and size. Adapted from [60].

scanning a large set of tape lift samples without human intervention. The development of a dedicated computer vision algorithm to analyse the resulting images could result in fully automated trace finding on collected trace samples. By reducing analysis costs, more trace samples can be analysed, allowing trace investigation to be applied more often and more rigorously.

However, the development of a computer vision microtrace recognition model is impeded by the required labelling to train such a model. State-of-the-art image recognition is mainly based on deep learning and relies on a large database of annotated training images [36, 61, 79, 85]. This is not available for trace recognition in tape lift scans and is expensive to obtain [16].

This thesis investigates the potential of self-supervised learning (SSL) to alleviate this issue. Recent research demonstrated that SSL allows for label-efficient learning for natural image classification by leveraging training on unlabelled data [13, 14, 28]. As collecting large amounts of unlabelled data is easy with respect to labelling data, self-supervised learning has seen increased interest in recent years. However, as self-supervised learning is relatively new, its applications in forensics and digital microscopy are still understudied.

1-1 Outline

In this chapter, the motivation and context of this thesis are given. In Chapter 2, relevant background on forensic microscopy, deep learning and self-supervised learning is discussed. Subsequently, Chapter 3 describes the main findings of this thesis in the form of a manuscript. Chapter 4 concludes the thesis with a summary of the main findings and recommendations for further research.

Chapter 2

Background

This chapter discusses the relevant background for the manuscript of Chapter 3. First, Section 2-1 introduces the automated microscope that was used to record the microtrace images regarded in this thesis. Then, Section 2-2 gives an introduction to deep learning image recognition models. Finally, Section 2-3 describes the concept of self-supervised learning (SSL) together with the SSL framework that is researched in the manuscript.

2-1 Trace analysis with automated microscopy

In the European project *Shuttle* [70], a digital automated microscope was developed that can scan microtrace samples in the form of tape lifts in high resolution in multiple microscopy imaging modes. This allows the digitisation of trace samples and paves the way for the application of computer vision models for automated trace recognition.

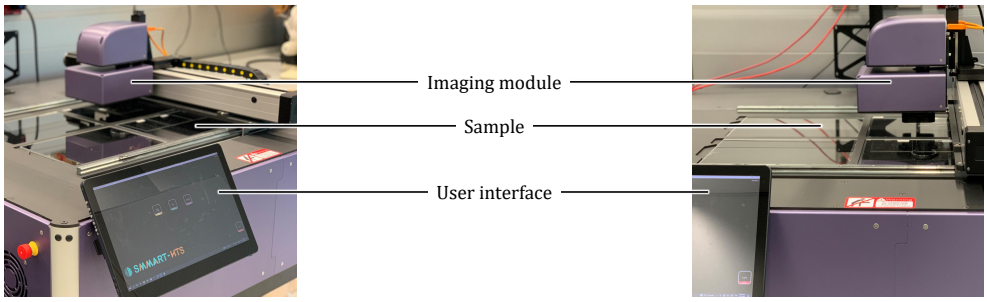


Figure 2-1: Photograph of automated microscope developed in [70]. The scans in this thesis are captured with this device.

The device consists of an imaging module that is able to move with respect to the sample (see Figure 2-1). By capturing neighbouring frames, the device can scan an area of four

A4 sheets without manual intervention. The primary imaging mode is transmission microscopy. Although the device also captures other imaging modalities (fluorescence, polarisation and reflection), the data of these modes were of poor imaging quality at the time of writing this thesis. Moreover, the exact image formation is proprietary [3]. Therefore, these imaging modes were not regarded in the thesis.

The device captures images of the samples with transmission imaging in a resolution of approximately 1 µm per pixel. For a tape of 80×80 mm, 999 images are captured (27×37 frames of 2920×2160 pixels). These images are joined together to form a full-size image of the microscopy sample. This technique is also known as *whole slide imaging (WSI)* or *virtual microscopy*. Although the application of WSI in forensic microtrace investigation is new, significant developments have been made by applications in pathology [29]. These developments include the application of deep learning computer vision models [25, 35, 42] and the developments of file formats and viewing software to efficiently store and share WSI scans [5]. A tool was developed in this thesis to convert the file format used by the automated microscope [70] into the open-source pyramidal `.tiff` image format to leverage these developments. This allows the scans to be viewed and annotated with software originally intended for pathology, such as [5].

The basic image formation process for transmission microscopy is visualised in Figure 2-2. The sample is illuminated from the opposite side of the camera module. Through light absorption, the objects appear dark across a bright background in the recorded image. The image is formed as follows [71]. The light source on the bottom of the sample emits approximately white light. After passing through the diaphragm, the condenser lens focuses the light on the sample. When the light passes through the sample, a portion of the light will be absorbed by the objects in the sample. The objective lens, which can also be implemented via a combination of lenses, magnifies the image and focuses the light on the imaging sensor. In the *Shuttle* automated microscope, the height of the imaging module containing both the objective lens and camera sensor is adapted to re-focus the image throughout the sample. This is required as the thickness of objects on the sample can vary. The camera sensor captures and digitises the optical image into pixel readings.

An example image recorded with the device is given in Figure 2-3. The sample is composed of a variety of traces scattered across a transparent tape of 80×80 mm. The traces absorb light, while the absorption in the tape is negligible. Therefore, the Figure shows a bright background with dark objects. The intensity of the transmitted light can be determined with the law of Beer-Lambert [78]:

$$I_1(\lambda) = I_0(\lambda) e^{-\varepsilon(\lambda) \cdot d}. \tag{2-1}$$

Here, λ denotes a certain wavelength, $I_1(\lambda), I_0(\lambda)$ the transmitted and incident light at said wavelength respectively, ε a material parameter and d the length of the light path throughout the material. For a compound system, the Beer-Lambert law yields

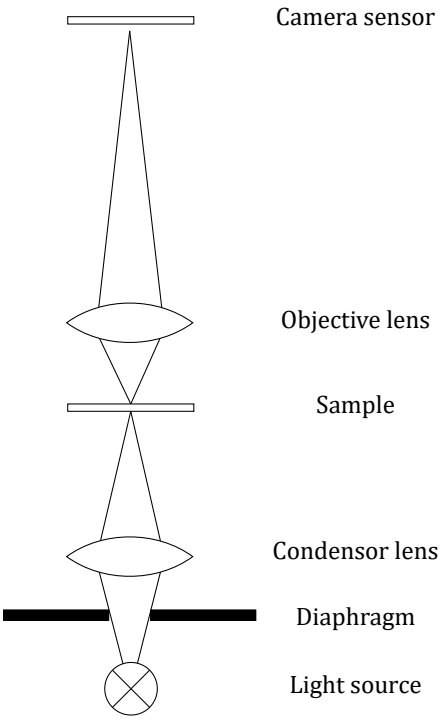


Figure 2-2: Simplified illustration of image formation.

$I_2 = A_1 I_1 = A_1 A_0 I_0$. Here, A denotes the absorption factor $e^{-\varepsilon(\lambda)\cdot d}$, which is bound between 0 and 1.

The absorption of light causes $I_1 \leq I_0$. However, light can be scattered in the neighbourhood of objects and artefacts such as air bubbles [71]. This can result in local concentrations of light yielding the rare occasion $I_1 > I_0$ for some pixels of the scanned image.

Ideally, the incident light is constant throughout the trace sample. Due to inhomogeneity in illumination, this is generally not the case [20, 78]. An example of this is given in Figure 2-4a. Here, a stiched-together overview image of a tape lift sample captured with 999 neighbouring frames is shown. The left side of each frame is imaged slightly darker than the right side, causing the individual frames to be clearly visible. This degrades the visual quality and complicates downstream tasks such as segmentation of regions of interest [55].

In Appendix A, a shading filter is designed. This filter is based on a linear approximation of image formation and uses median filtering to determine an approximation of a calibration image. The result of this filter is shown in Figure 2-4b. Here, it can be seen that the filter homogenises the background and successfully removes the discontinuities at the edge of each frame.

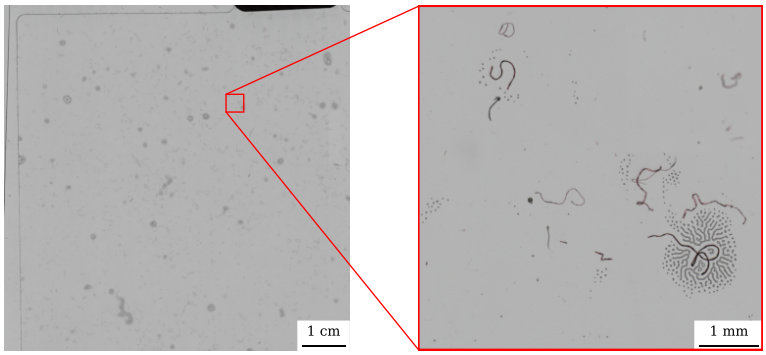
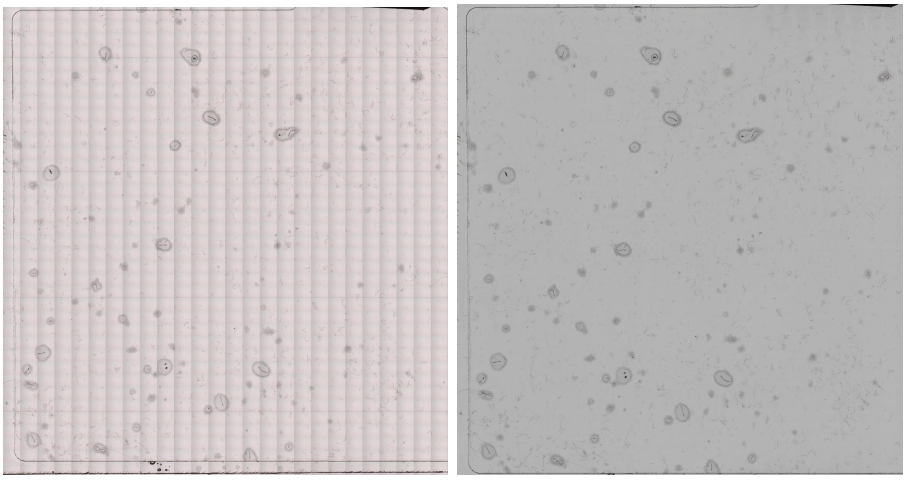


Figure 2-3: Example of microtrace image captured in transmission imaging mode. A bright background with dark outlines of objects can be seen. The zoomed-in view shows red cotton fibres. The fibre at the bottom right of the zoomed-in view is accompanied by air bubbles.



(a) Original version. **(b)** Corrected version.

Figure 2-4: Result of shading correction.

2-2 Deep learning for image recognition

Recognising objects such as microtraces in images is a sub-field of computer vision. While earlier research was mainly based on hand-crafted feature extractors, modern computer vision models use deep learning and directly train on the pixel values of images via artificial neural networks [10, 21]. This allows the model to learn more diverse and more complex patterns. In various fields and computer vision tasks, deep learning approaches have therefore outperformed traditional approaches in accuracy [45].

Artificial neural networks

In the most general sense, an artificial neural network consists of a set of layers with sets of weights [27]. Each neuron l in a layer k takes a vector of input values $Z_{k,l} \in \mathbb{R}^{N_m \times 1}$, has a set of weights $\theta_{k,l} \in \mathbb{R}^{1 \times N_m + 1}$ and outputs a scalar value $a_{k,l}$ based on a nonlinear activation function σ [6]:

$$a_{k,l} = \sigma \left(\theta_{k,l} \begin{bmatrix} 1 \\ Z_{k,j} \end{bmatrix} \right). \tag{2-2}$$

The first weight $\theta_{k,l,0}$ is referred to as bias, as it is independent of the inputs of the neuron. The inputs Z can be based on the input data X_i or the output values of earlier activated neurons. By stacking layers and using the outputs of neurons as input for deeper neurons, increasingly abstract features can be recognised from the input data [6]. The final layer outputs the prediction of the model:

$$\hat{Y} = f_\theta(X). \tag{2-3}$$

Here, θ denotes the network weights, while f denotes the network architecture that defines the connection and activation of each neuron. The weights are trained by minimising a loss function $\mathcal{L}$. In supervised training, a set of desired output values Y is available. In this case, the loss is often a function of the distance between $\hat{Y}$ and Y [27]:

$$\theta^* = \underset{\theta}{\operatorname{argmin}} \, \mathcal{L}(|| \, f_\theta(X) - Y \, ||). \tag{2-4}$$

Here, θ^* denotes the optimal weights. For the optimisation itself, Stochastic Gradient Descent (SGD) is a common choice. This relates to the general optimisation algorithm Gradient Descent. With Gradient Descent, each optimation step k updates the parameters based on the gradient of the cost function [27]:

$$\theta_{k+1} = \theta_k - \eta \nabla_\theta C(\theta_k). \tag{2-5}$$

Here, η denotes the learning rate. In SGD, the gradient is approximated through mini-batches of data points to keep calculations tractable. For example, when a batch size of 64 is used, the optimisation process calculates the loss function and its gradient for 64 data points, performs an update step and then moves on to the next 64 data points. As the calculated gradient for a small batch size is noisy, smaller learning rates are commonly chosen for small batch sizes [12, 28].

The most simple architecture is to fully connect all neurons in each layer to all neurons in the previous layer. At least three subsequent fully connected layers form an architecture called an multilayer perceptron (MLP) [6]. This is visualised in Figure 2-5.

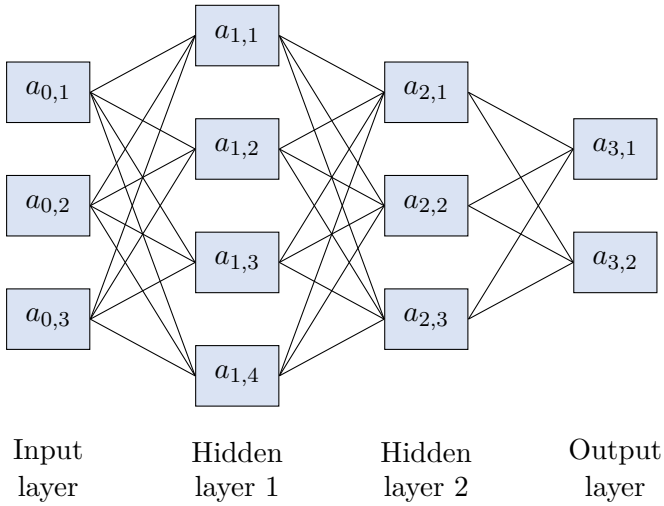


Figure 2-5: Illustration of a multilayer perceptron (MLP) with two hidden layers. The blocks visualise neurons, the lines present the weights. Each neuron is fully connected to each neuron of the layer before, resulting in a total of 30 weights (when no bias is regarded).

Convolutional neural networks

When a fully connected layer is applied to an image, each weight is trained on one colour channel of one pixel. This results in an immense amount of trainable weights and would not regard the inherent 2D structure of an image.

Therefore, image recognition models often employ 2D convolutional layers [27, 32, 39]. In these layers, a kernel with learnable weights is slid across the input through convolution. This allows the weights to learn features independent of where they are located in the input [27]. For example, a kernel might be dedicated to recognising edges or colour blobs throughout the image. The resulting activation map of the layer then contains information on where these features are located, which can subsequently be used to recognise more complex features such as shapes and textures. For a layer k, the activation map A_k of a 2D convolutional layer is generated with a kernel K as [27]:

$$A_k(l_1, l_2) = (K * Z_k)(l_1, l_2) = \sum_m \sum_n K(m, n) Z_k(l_1 - m, l_2 - n). \tag{2-6}$$

The convolution can also be implemented with a *stride* s larger than one. In this case, the kernel is slid across the previous layer with steps of size s. This results in a downsampled output map and is therefore used to reduce computational cost [27].

A convolutional layer can also be implemented with *dilation*. Here, zeros are inserted into the kernel to increase the size of the activation region, without requiring a larger number of trainable weights. To illustrate this, Figure 2-6 compares a standard convolution, a

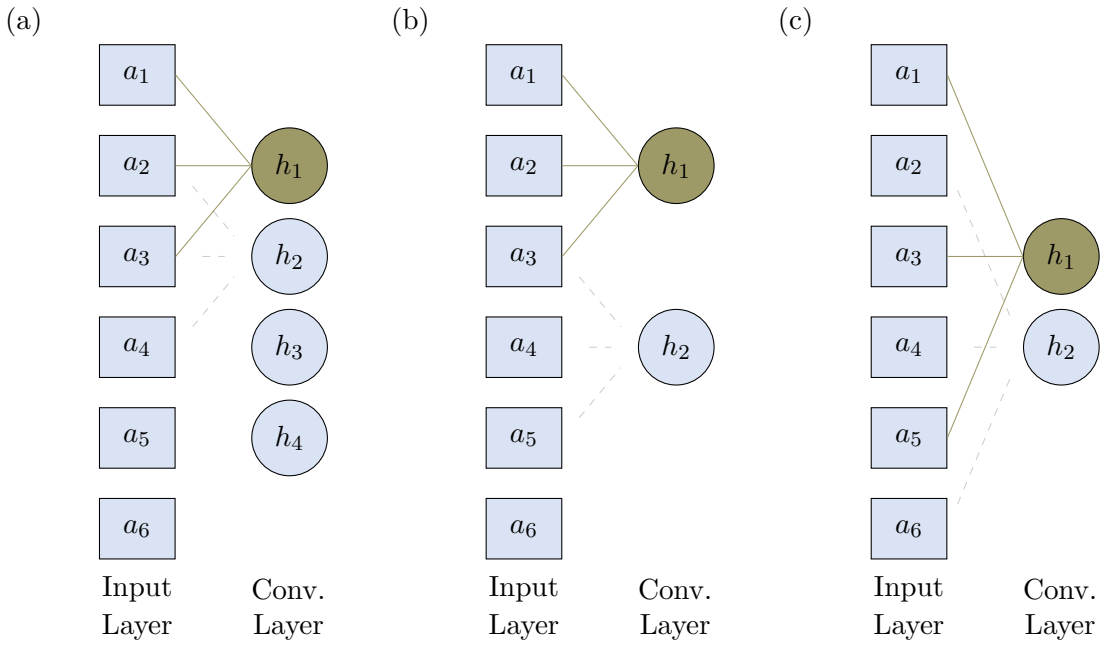


Figure 2-6: Illustration of 1-dimensional convolutional layer with a kernel size of 3. (a) Regular convolution with stride 1 and no dilation. (b) Strided convolution with stride 2. (c) Dilated convolution.

strided convolution and a dilated convolution for a 1-dimensional layer. It can be seen that the dilated convolution covers a wider region while still requiring only three weights. In image processing, the increased field of view of the activation helps in capturing contextual information [10].

Convolutional layers are the foundation of convolutional neural networks, which are often used as feature extractors. They aim to reduce the raw input pixel values into a feature space of smaller dimensionality that embeds the required information for the downstream task. This can be for example classification.

When stacking a large number of layers in a neural network, the neurons in the early layers are increasingly difficult to optimise. For these neurons, the gradient has to flow through all subsequent (non-linear) operations [32]. This can cause the optimisation to get stuck in a local minimum of low accuracy. Therefore, residual blocks are proposed in [32]. These blocks introduce a skip-connection that sums the input a_{in} of a neural network block $\mathcal{F}_\theta$ with the output a_{out} as shown in Figure 2-7. This allows the gradient of earlier layers to flow directly to the later layers, bypassing $\mathcal{F}_\theta$. This results in a smoother optimisation landscape, making optimisation easier [41].

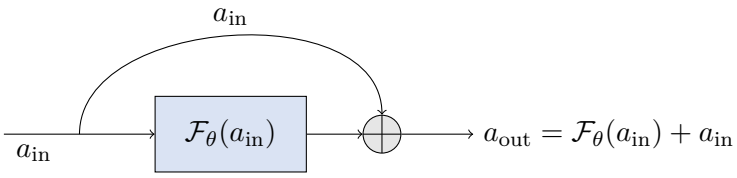


Figure 2-7: Illustration of residual block proposed in [32].

A residual network with 50 layers (ResNet-50) is one of the most common architectures for computer vision at the time of writing [2, 33, 83]. The main components are residual blocks, convolutional layers and downsampling layers.

For image classification, the extracted feature maps are commonly further processed with an MLP. Each neuron in the output layer corresponds to one of the classes. Thus, this layer should have the same number of neurons as the number of classes N in the training data. The neuron with the highest activation denotes the class that the network regards as most likely for that image.

Often, a soft-max cross-entropy loss function is used. First, a softmax function normalises the activations of the last layer a_k such that they can be interpreted as a categorical probability distribution with $\sum_l p_l = 1, p_l \geq 0$ for each class/neuron $l = 1, \cdots, N$ as [27, 62]:

$$p_l = \text{softmax}(a_k)_l = \frac{\exp(a_{k,l})}{\sum_{j=1}^{N} \exp a_{k,k}}. \tag{2-7}$$

Considering the desired output probability distribution Y_i and predicted probability distribution $\hat{Y}_i = \begin{bmatrix} p_1 \cdots p_N \end{bmatrix}^T$ for data points $i = 1 \cdots N_i$, the cross entropy is defined with the negative log-likelihood as [6, 38]:

$$\mathcal{L}\left(f_\theta(X), Y\right) = -\sum_{i=1}^{N_i} \sum_{l=1}^{N_A} \left(Y_i(l) \ln \hat{Y}_i(l) + (1 - Y_i(l)) \ln \left(1 - \hat{Y}_i(l)\right)\right). \tag{2-8}$$

When the annotations do not regard uncertainty, the expert annotation is formatted as a vector with length N with a one indicating the correct class and zeros for the other classes. This is called one-hot encoding. In this case, the cost function reduces to [27]:

$$\mathcal{L}(f_\theta(X), Y) = -\sum_{i=1}^{N_i} \sum_{l=1}^{N_A} Y_i(l) \ln \hat{Y}_i(l). \tag{2-9}$$

The advantage of using the softmax cross entropy over for example a loss function based on accuracy is its differentiability. For instance, a binary classification for one point would result in an accuracy of either 0 or 1, while the cross entropy outputs a continuous range.

Semantic segmentation

In the computer vision task of semantic segmentation, each pixel is classified separately. This allows for precise localisation of each class in the image. Here, common practice is to employ additional convolutional layers instead of an MLP in the final prediction layers [62, 67].

These convolutional layers build classifications on the feature map via 1×1 convolutions and upsample them to yield a pixel-wise prediction map. For upsampling, transpose convolutional layers are used. Transpose convolutional layers can be understood as a standard convolutional layer with a fractional stride of $0 < s < 1$ on the input map. Thus, the kernel takes steps smaller than one in its convolution operation and outputs a value for each step, yielding an activation map larger than the input map.

When the input image is first reduced to a small feature map and then again upsampled to a high-resolution prediction map, fine-grained details may get lost. Therefore, [67] proposes to use an additional set of skip-connections. With these skip-connections, predictions are made on feature maps of small dimensionality as well as on feature maps of larger dimensionality (see Figure 2-8). By combining these predictions, the prediction map allows more fine-grained details.

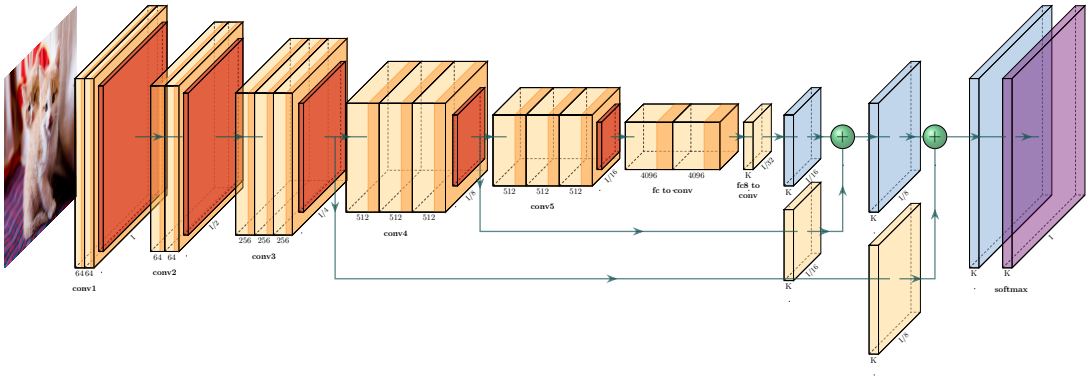


Figure 2-8: FCN-8s architecture [67] for semantic segmentation. The yellow, orange and blue blocks visualise convolutional, downsampling and transpose convolutional layers respectively. The purple block presents the mask containing the predicted labels. Source: [34]

Another method to avoid to loss of fine-grained details is to use higher-resolution feature maps. For example, the stride of a convolutional layer can be decreased. This will cause the output map of the layer to be larger. However, when the kernel size of a subsequent convolutional layer is kept the same, the kernel of this next layer is activated by a smaller region of the input image. Therefore, dilated convolutions (see Figure 2-6c) are often used in recent semantic segmentation models [10, 11].

Commonly in semantic segmentation, the loss function is the sum of the per-pixel softmax cross entropy loss, while the model evaluated with the mean Intersection over Union (mIoU) [42, 49, 50, 84, 87]. The mean Intersection over Union (mIoU) is defined as:

$$\text{IoU}_A = \frac{\text{area}(y_A \cap \hat{y}_A)}{\text{area}(y_A \cup \hat{y}_A}), \tag{2-10}$$

$$\text{mIoU} = \frac{1}{n_c} \sum_A \text{IoU}_A. \tag{2-11}$$

In the top equation, the number of pixels both predicted and annotated as class A (intersection) is divided by the number of pixels either predicted or annotated as class A (union). Thus, the perfect score is 1 (no false positives and no false negatives) and the worst score is 0 (no true positives). In the bottom equation, the mIoU is calculated as the mean of the over the n_c non-background classes.

The advantage of using the mIoU for evaluation over the pixel accuracy is that it is less sensitive to class imbalance. When 80% of a picture is background and a model predicts 100% background, the pixel accuracy is 80% although none of the objects is classified properly. As microtraces only span a small area in tape lift scans, using the mIoU is a more suitable metric to quantify the performance of microtrace segmentation.

2-3 Pretraining with contrastive self-supervised learning

Recent research has shown that pretraining a feature extractor with self-supervised learning (SSL) allows learning useful features for classification without requiring explicit training labels [14, 28]. This allows label-efficient learning, in which the required number of labels to achieve accurate predictions is decreased [13].

The objective of SSL is to train a model to differentiate relevant features from irrelevant ones without requiring a supervisory signal about the exact meaning of those features. To illustrate this concept, consider a scenario from the field of natural language processing. When humans learn a new language, a substantial portion of their learning process involves reading texts without constantly consulting a dictionary for each word. Instead, they acquire knowledge about word relationships and sentence structures, even in the absence of a complete explicit understanding of the text. In this context, the dictionary serves as a supervisory signal, while the unknown texts function as unsupervised data. The concept of SSL aims to replicate this form of learning by generating a puzzle based on unsupervised data. In natural language processing, a typical puzzle might involve removing a word from a text and determining which word was omitted [19]. Although this puzzle does not require explicit knowledge of the text, it requires a form of understanding of sentence structures and relations. Therefore, training on this puzzle can help with learning the language.

Although other types of methods have been proposed for SSL in computer vision [22, 26, 48], recent contrastive methods have shown the most powerful performance in learning useful features for classifications [1, 43, 77].

The core concept of contrastive SSL models is to create two different augmentations of the same image and train the model to recognise that the two augmented images correspond to the same object(s) [12]. This training objective aims to induce invariance to the applied augmentations. To compare the model's interpretation of each image, feature vectors are used.

Formally, two augmentations (V_i, V_i') are created from the input image X_i. The weights of the model are optimised to maximise the similarity between the feature representation of both augmentations. A simple cost function without regularisation would be:

$$\mathcal{L}_\theta = || \ f_\theta(V_i) - f_\theta(V_i') \ || \ . \tag{2-12}$$

However, a fundamental issue arises. Namely, the above minimisation results in a perfect cost value of zero when the network outputs a constant feature representation $f_\theta(V_i) = c$ for all inputs V_i. In this case, the learned feature representation is independent of the input and constant for each input image X_i. Therefore, it is useless for further usage in the downstream task. The issue of learning this identity mapping is called model collapse in SSL [14]. Various methods have been proposed to try to avoid it.

An intuitive approach is to not only use augmentations of the same image (positive pairs) but also to include comparison with other images (negative pairs). This is illustrated in Figure 2-9. This was proposed in the SSL framework *SimCLR*, which demonstrated competitive accuracies on ImageNet classification with respect to earlier SSL models [12]. By introducing a large cost for $f_\theta(V_1) \approx f_\theta(V_2)$ for a negative pair consisting of two different images V_1, V_2, it avoids model collapse. However, sampling a large number of negative pairs results in requiring a large batch size [12], which is computationally demanding.

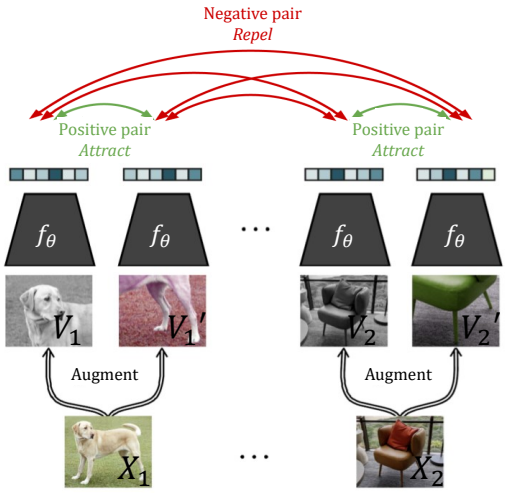


Figure 2-9: Positive and negative pairs in contrastive SSL. Source: adapted from [12, 15].

2-3-1 BYOL

Recent contrastive SSL models are able to circumvent the usage of negative pairs [14, 28, 28, 57]. In particular, BYOL ('Bootstrap Your Own Latent') is proposed in [28]. This framework uses solely positive pairs and shows powerful performance in the classification and segmentation of natural images. By refraining from using negative pairs, it maintains performance for smaller batch sizes [28]. This allows the model to be trained with smaller memory requirements.

BYOL is illustrated in Figure 2-10. The model is composed of two separate networks: an online network with weights θ and a target network with weights ξ. The online network consists of a feature extractor, a projector and a predictor. The target network on the other hand only consists of a feature extractor and projector. These have the same architecture as the online feature extractor and online projector but have different weights.

First, two augmented versions V_i, V_i' are generated for an input image. Both networks receive one of the versions as input. Then, the online network aims to predict the output of the target network. As this requires understanding what can be seen in the image, this results in training the feature extractor to generate useful image representations.

The target network on the other hand receives an exponential moving average of the weights of the online feature extractor and online projector. Hereby, it produces stable targets to optimise the online network on. While recent research has shown that the target network and online network can have identical weights [14], the exponential moving average improves performance by providing smooth changes in the target representations [28].

The asymmetric network structure of BYOL is found to be crucial in avoiding model collapse [74]. Although the model is theoretically still susceptible to collapse [28], it is empirically found that the asymmetric updating process prevents convergence into an identity mapping [4, 28, 75]. In related work [14], it is concluded that it is unlikely that the output of a randomly initialised network has little dependence on the input and that an optimiser optimising only a single part of an asymmetrically updated structure can find a trajectory that converges to an identity mapping.

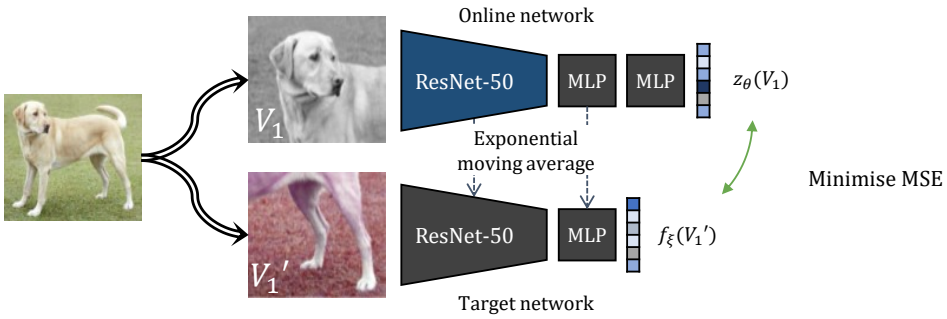


Figure 2-10: Illustration of the contrastive SSL model BYOL [28].

Chapter 3

Reducing Manual Labour in Forensic Microtrace Recognition with Deep Learning

In this chapter, the main findings of this thesis are presented in the form of a manuscript formatted for publication in Nature Scientific Reports. Supplementary materials to the manuscript can be found on pages 32 — 53.

Reducing Manual Labour in Forensic Microtrace Recognition with Deep Learning

Gerben Rijpkema[1], Dylan Kalisvaart[1], Serafim Korovin[1], Daniel Spengler[1], Anna Pals[2], Jaap van der Weerd[2*], Carlas Smith[1*]

Abstract

Forensic microtrace investigation relies on a time- and labour-intensive process of manually analysing samples via microscopy. To aid forensic experts in their investigations, an image recognition model for microtrace localisation and classification is needed. In this work, we show the trace recognition accuracy that can be achieved by analysing images captured with automated microscopy through deep learning. We localise and classify fibres, hairs, skin, glass and sand in microscopy scans through pixel-wise classification of tape-lift samples. As deep learning requires extensive amounts of annotated training data, we additionally investigate pretraining to minimise the required annotation workload. We compare ImageNet pretraining, pretraining with self-supervised learning and a sequential application of these approaches. We find that pretrained models are able to reduce the required annotated data twofold compared to models trained from scratch, while retaining the prediction accuracy. While our ImageNet-pretrained models outperform our self-supervised-pretrained models, we achieve the highest accuracy by combining the two approaches. With this, we can recognise traces with a mean intersection over union of 0.56 when training on only 2.2 dm^2 of annotated tape lift scans. Our model is therefore the method of choice for automatic analysis of large forensic microtrace scans.

I. Introduction

The presence and transfer of microtraces such as hairs, skin cells and fibres are used in forensic investigations to reconstruct a crime by providing information on people, objects, locations and actions [59]. One of the used methods to recover these traces from areas of interest is one-to-one tape lifting. Here, a large number of transparent tapes is used to fully cover a body, surface or object to thoroughly recover traces and preserve their locations [17, 18].

Laboratory investigations on one-to-one tape lifts are costly, as tape lifts may be large. Taping a human body results in roughly 1.5 m^2 of tape [18]. These tapes generally contain many small traces and require microscopic investigation by well-trained microscopists [60, 65, 68]. In most cases, trace investigation is therefore limited to selected traces and regions of interest [18, 51] or not applied at all [56]. This results in unexploited forensic potential.

Automation of trace investigation may reduce manual labour and cost and hence improve the effectivity of trace evidence investigations. A recent step in automation has been made with the development of automated microscopy systems within the *Shuttle* project [70] to capture digital scans of tape-lifted samples. However, a key challenge that remains is recognising microtraces in the captured scans. In particular, an image recognition model capable of classifying and localising traces would enhance the investigation process by enabling rapid processing of microtrace scans.

In this work, we develop a deep learning method for pixel-wise classification (semantic segmentation) of traces in microscopy scans and show its ability to precisely localise and classify traces on tape-lifted microtrace samples. This model compiles overviews that show the identity and distribution of some of the most useful microtraces. These overviews assist experts in deciding on further investigation procedures.

Deep learning image recognition relies on training a neural network to derive meaningful information directly from the pixel values of input im-

[1]Delft Center for Systems and Control, Delft University of Technology, Delft, The Netherlands, [2]Netherlands Forensic Institute (NFI), The Hague, The Netherlands, *these authors contributed equally

ages. This allows for learning complex patterns and relationships within images. As a result, deep learning has achieved remarkable success in visual tasks such as image classification and semantic segmentation [16, 32, 39]. Typically, the network is trained with a large database of annotated images [8, 16, 32, 36, 79]. Compiling this database requires extensive manual labour, as acquired images need to be labelled by experts. As a variety of traces can be encountered with tape lifts and the number of traces per sample is low [72], a large number of tapes has to be annotated by experts. Moreover, the cost of labelling training images for segmentation is high [16] as each pixel in the training data has to be annotated.

Therefore, we investigate how annotation workload can be kept minimal through pretraining. With pretraining, the network is optimised to extract visual features from images that are useful for recognising microtraces without requiring annotated microtrace images. In particular, we explore three pretraining strategies and evaluate their ability to decrease the required amount of annotations needed for trace recognition.

Firstly, we test ImageNet classification pretraining [37, 64]. Here, a neural network is pretrained through classification of everyday photographs in the large scale ImageNet database [64] containing over 1 million images. In this way, it learns to recognise visual features important for human observers, such as the colour, shape, and texture of objects in the input images. It has been shown that the learned visual features remain useful when applied to other visual tasks, thereby benefitting convergence speed and accuracy [85]. Furthermore, ImageNet pretrained models are widely available, therefore requiring no additional computational efforts in pretraining [53]. However, the benefits of ImageNet pretraining diminish for tasks dissimilar to the classification of everyday photographs, as the learned visual features can be sub-optimal for other domains [37, 69].

Therefore, we test a second strategy of pretraining with self-supervised learning (SSL) on unannotated microtrace images. We use the framework proposed in [28] to optimise the network weights such that similar features are extracted for two artificial modifications (*augmentations*) of an unannotated microtrace image. We create two versions via cropping, recolouring and rotating and minimise the distance between the representations of both versions. By learning the similarity of the content of the images, the network is trained to extract useful features for microtrace classification without requiring an expert annotation for each image.

Thirdly, we investigate a combined pretraining approach in which we first use ImageNet-pretraining to learn general visual features and subsequently apply SSL pretraining to further specialise in microtrace-specific features. This approach follows [58].

Figure 1 presents a schematic representation of our deep neural network for automated trace classification. In Figure 1a, we classify pixels in microscopy scans as either fibre, hair, skin, glass, sand or background. The network is composed of a ResNet-50 [32] feature extractor that embeds the relevant information of the input image into a feature representation. This feature representation is subsequently used by a fully convolutional [67] pixel classifier to assign a predicted label for each pixel (semantic segmentation). The network can be trained from scratch by initialising the weights with random parameters and training only via annotated microtrace images. Alternatively, we use an ImageNet-pretrained feature extractor (see Figure 1b) or pretrain the feature extractor with SSL on unannotated microtrace images (see Figure 1c). In Figure 1d, we show our method for extracting and augmenting training images from microtrace scans.

We evaluate the trace recognition ability of the model with the mean Intersection over Union (mIoU) to measure the overlap between the predicted traces and the annotated traces (see Subsection IV.F). By combining ImageNet pretraining with SSL pretraining, we are able to recognise hairs, fibres, skin and glass with an mIoU of 0.56 while only being trained on 2.2 dm^2 of annotated tape lift scans. A model trained from scratch on the same amount of annotated data achieves an mIoU of 0.34. We can outperform this with pretraining even if we reduce the available amount of annotated data four-fold to 0.6 dm^2, with which we achieve an mIoU of 0.36. Our results provide in-

sight into the required amount of labelling to recognise microtraces in microscopy scans. Furthermore, by facilitating trace analysis with a graphical user interface through [5], we enable automatic tape lift analysis to aid forensic experts in their investigation.

II. Results

We first present the microtrace recognition of the model with which we achieve the highest mIoU in Subsection II.A. This model was sequentially ImageNet pretrained, SSL pretrained on roughly 1 m^2 of unannotated tape area and then trained on our annotated dataset of 0.022 m^2 of tape area. In II.B, we report our results on the benefit of pretraining for the required amount of labelled training images. Then, we discuss our results on self-supervised learning in II.C. Finally, we provide insight into the model and its key elements by removing parts of the pipeline and reporting the ablation results in II.D.

A. Microtrace recognition

We achieve the highest mIoU by using combined ImageNet and SSL pretraining and training the trace predictions on all annotated data. Although our annotated dataset is relatively small and considers only a tape lift area of 2.2 dm^2, we find an overall mIoU of 0.56.

Generating predictions for a microtrace tape of 80×80 mm takes approximately 30 seconds using our hardware (NVIDIA RTX3090 GPU and Intel i910900X CPU). This is a fraction of the time required for a microscopist to generate an overview of traces on the tape. Note that the bottleneck in our method is scanning the tape with automated microscopy, which is found to take 40 minutes with the device developed in [70].

Supplementary Figure S1 shows the confusion matrix of the model together with precision and recall analysis. It can be seen that the pixel-wise precision exceeds 60% for each class, indicating that at least 60% of the pixels marked as a certain class are marked as that class by the expert annotator as well. Except for the class Skin, the recall exceeds 60% as well, indicating that at least 60% of the pixels are marked as a certain class by both the expert and the model.

To visually assess the quality of the model predictions, we provide examples of model predictions in Figure 2 and Supplementary Figure S2. Here, image patches are shown together with expert annotation and model prediction. We provide IoU values for classes for which the union of predicted and annotated pixels spans at least 1% of the image patch.

Figure 2 shows correctly recognised traces as well as incorrect predictions. Hairs, sand and glass is correctly detected. Air bubble patterns can be seen in the first three images as well as in the sixth and last image. These are caused by air being trapped under the tape. It can be seen that the model correctly refrained from these patterns as traces. The second column shows an incorrectly classified trace. Whereas the expert annotated a fibre, the model prediction fluctuates between hair and fibre. The last column shows a fibre at the top that is partly missed by the model.

In Supplementary Figure S2c, we show additional image patches on which the model performs poorly. These images were randomly selected from predictions where a class IoU of less than 0.4 was encountered.

In the seventh column of Figure S2c, a cluster of skin cells in the form of dandruff can be seen. Due to the thickness, the transmitted light is low and the trace is opaque. Therefore, the trace is visually similar to a sand particle (shown in the fifth column of Figure 2), causing a misclassification by the model. To improve the accuracy for these traces, alternative microscopy modes such as reflection imaging should be incorporated.

B. Pretraining for label-efficient learning

In Figure 3, we report the results for training with different amounts of labelled data. Here, we vary the annotated tape-lift area used in training the trace classifications. Since we encounter low class diversity when training on areas of 0.1, 0.3 and 0.6 dm^2, we average over 5, 3 and 2 training runs

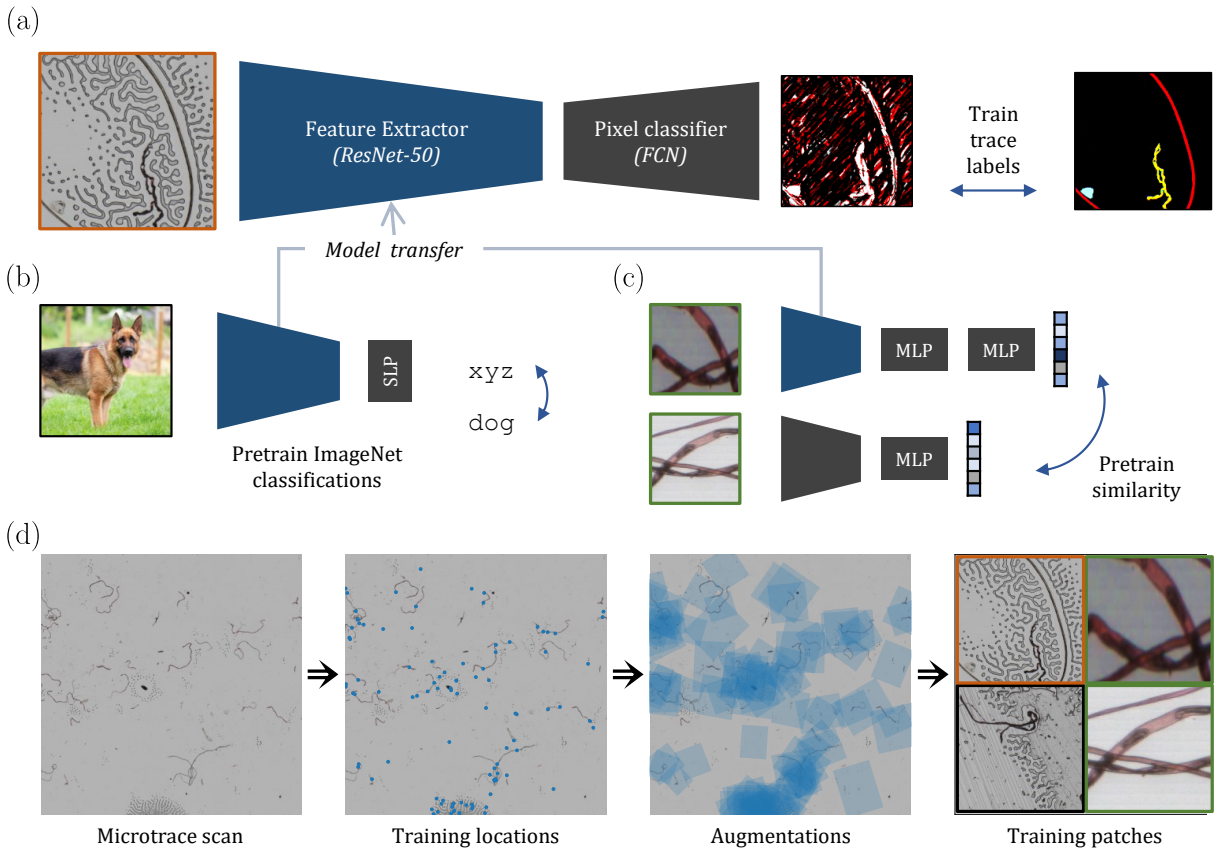


Figure 1: Schematic overview of deep neural network for automated microtrace classification. (a) Training trace predictions with expert annotations. A residual network with 50 layers (ResNet-50) [32] feature extractor is used in a fully convolutional network (FCN) [67] structure for pixel-wise classification. The network aims to find the relationship between the microtrace images (left) and the corresponding labels provided by experts (right) by minimising the cross-entropy loss between the predicted and annotated labels. The microtrace image in the figure shows a hair, a fibre and air bubbles. (b) Pretraining the feature extractor with the self-supervised learning (SSL) model BYOL [28] on pairs of unannotated images. Here, the weights of the feature extractor are adapted such that pairs of similar images yield similar features. The architecture is composed of two ResNet-50 networks and a set of multi-layer perceptrons (MLPs). (c) ImageNet pretraining of feature extractor through classification of annotated everyday photographs in the ImageNet database [64] with a single layer perceptron (SLP) classifier. (d) Extraction and augmentation of training images from tape-lift scans. For pretraining with SSL, image pairs are extracted. For training trace predictions, image patches are extracted together with their corresponding annotation mask.

respectively. We randomly select a tape lift area of said size for each run and train solely on image patches in that region (see Subsection IV.F).

For each of the experiments, the pretrained models result in higher mIoU values than the models trained from scratch. The ImageNet-pretrained models outperform the SSL-pretrained models. Combining ImageNet pretraining with additional SSL pretraining results in the highest mIoU values, although the benefit with respect to ImageNet pretraining only diminishes when annotated data ranges below 0.6 dm^2 of annotated tape.

When training on 0.6 dm^2 of annotated tape or more, all pretrained models outperform models trained from scratch using two times more annotated data. Moreover, combined ImageNet pretraining and SSL pretraining with 0.6 dm^2 of annotated tape results in a higher mIoU than training from scratch with four times more annotated data.

The mIoU (see Subsection IV.F) considers three

error modes. Firstly, microtrace pixels can be missed. Secondly, traces can be confused. Thirdly, traces can be incorrectly detected. In Supplementary Figure S3, we report the frequency of each of the error modes for our experiments. When training on 2.2 dm^2 of annotated tape, combined ImageNet and SSL pretraining results in a 40% increase in correctly classified trace pixels, a 27% decrease in missed trace pixels and a 56% decrease in false detections compared to training from scratch. The number of confused trace pixels remains similar (+1% increase).

C. Self-supervised pretraining

During SSL pretraining, we minimise the distance between the extracted features of two different augmentations of a similar image patch. We obtain these image pairs by selecting a point on a microtrace scan, extracting two images within a maximum distance d_m of this point and randomly augmenting each image (see Subsection IV.D). The results of the SSL experiments of Subsection II.B were obtained with $d_m = 0$ and a maximum magnification augmentation between 0.5× and 2× (sampled as described in Supplementary Table S2).

In Supplementary Figure S4a, we substantiate this choice by reporting results for different translation and magnification augmentation bounds. For this experiment, we train for 7 million data points instead of the 40 million data points that we use in our other experiments to decrease computational effort (see Figure S11 and Subsection IV.C). We train the pixel classifications on our full annotated dataset of 2.2 dm^2. With this amount of annotated training data, training from scratch results in an mIoU of 0.34 (see Figure 3b). Supplementary Figure S4a shows that the highest mIoU is achieved with $d_m = 0$ and magnification augmentation between 0.5× and 2×. Here, a mIoU of 0.42 is achieved. Choosing a zoom augmentation of lower intensity, namely between 0.67× and 1.5×, results in a sharp drop in mIoU. Here, we find an mIoU of 0.37, which translates into a 63% reduction of pretraining benefit. This follows from the fact that a light zoom augmentation and no translation results in two approximately equal views. In this case, training the feature extractor to yield similar representations for both views results in a trivial solution, resulting in a diminished benefit of pretraining. When a light zoom augmentation (between 0.67× and 2×) is chosen in combination with a translation augmentation of $d_m = 64$, the pretraining benefit returns (mIoU of 0.41, 88% of the highest pretraining benefit reported in the ta-

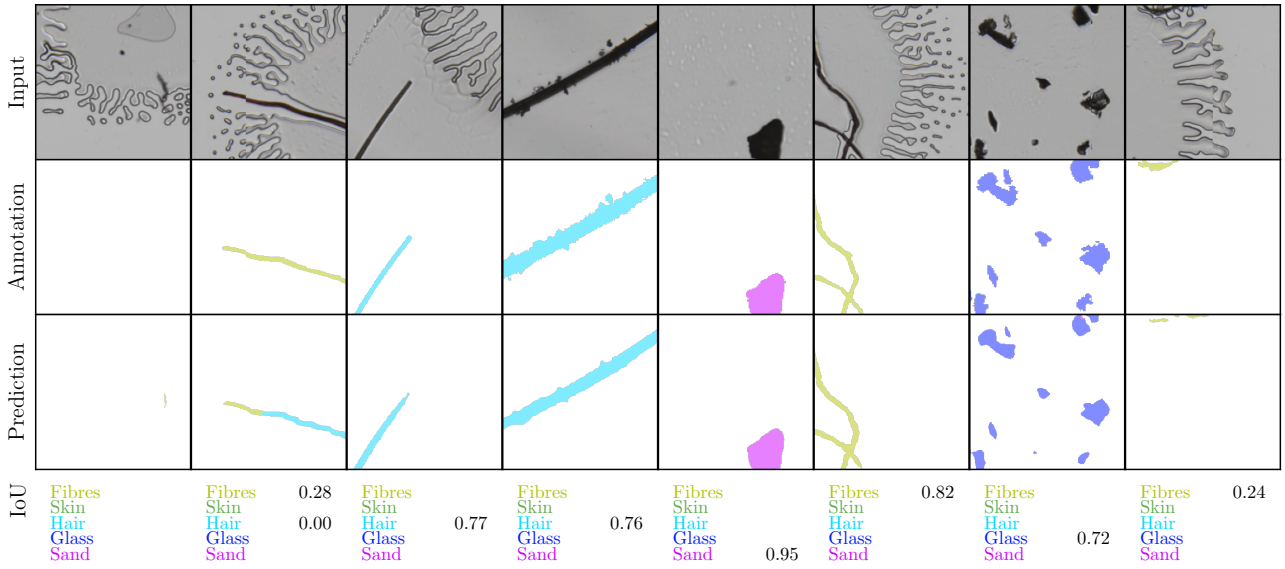


Figure 2: Comparison of model predictions with expert annotations for patches in the test set to visually assess model performance and corresponding Intersection over Union (IoU) values. The IoU values for classes that span less than 1% of the total image area in both the expert annotations and the predictions are omitted.

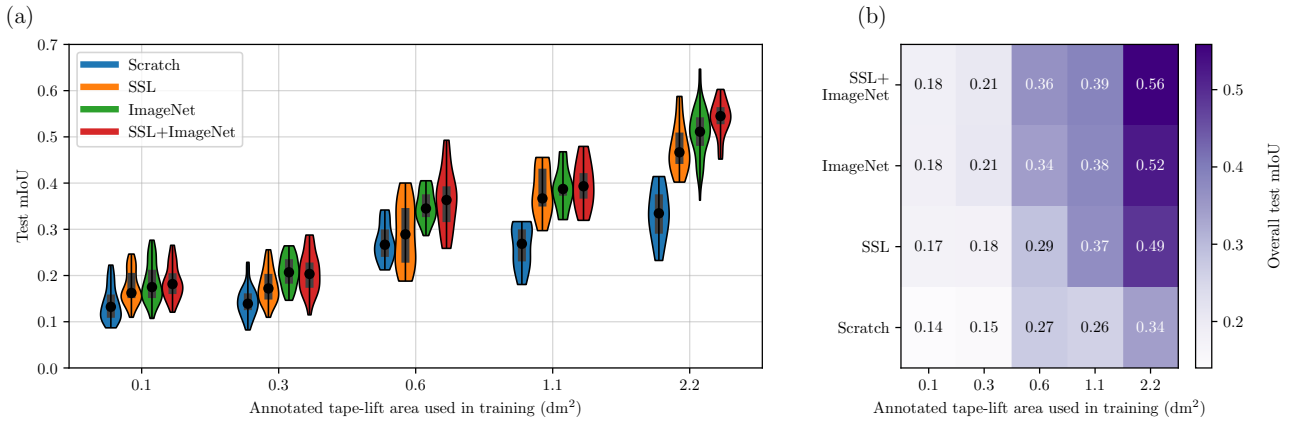


Figure 3: Benefit of pretraining for label-efficient learning. (a) Violin plot showing the distribution of mIoU over a 10-fold split in the test set. The dot presents the median, the inner bar the upper and lower quartile. The line presents the spread between the minimum and maximum. (b) Overall test mIoU per experiment.

ble).

Supplementary Figure S4b-f show the IoU benefit per class. When considering a heavy zoom augmentation (between 0.25× and 4×), the optimal value of d_m varies per class. While for hair and skin, increasing d_m from 0 to 256 µm results in +0.06 and +0.04 mIoU respectively, the IoU for glass decreases with -0.09. For sand and fibres, the difference in mIoU is negligible (±0.01). This is explained by the size of the traces. In Table 2, it is shown that the glass particles of our dataset have sizes in the order of $0.2 - 0.4$ mm. A translation of 256 µm and a 4× zoom can therefore cause the trace to move out the field of view for one of the images in the augmentation pair. In this case, the extracted features will be incorrectly optimised to be similar. On the other hand, hairs have lengths of more than 5 mm and skin is encountered in clusters, yielding a larger probability of encountering the trace in both views. In this case, the translation augmentation results in correctly learning the similarity of two views of the same trace. For these traces, the increased translation augmentation results in stronger feature diversity in the views. This increases the benefit of pretraining [73], resulting in the increased IoU.

D. Ablations and influence of hyperparameters

To gain a better understanding of the trace recognition model itself and its various elements, we perform ablation experiments. Here, we remove parts of our pipeline to investigate the influence on the mIoU of trace recognition.

First, we investigate our proposed image sampling approach involving thresholding (see Subsection IV.D and Figure 1d). We compare our approach to extracting image patches from scans with a uniform grid as shown in Supplementary Figure S5.

In Supplementary Figure S6a, it can be seen that after training on 400.000 image patches, our approach results in a 21% higher overall test mIoU compared to training with a uniform grid. When training with only 80.000 data points, corresponding to 4 full dataset iterations for the uniform approach, our thresholding approach results in a fourfold increase in mIoU.

We further investigate the sampled pixels in Supplementary Figure S6b and c. In Supplementary Figure S6b, it can be seen that the thresholding approach alters the class distribution in the sampled images. Without thresholding, the image area is sampled uniformly, resulting in a sampled class distribution that follows the overall class distribution of the dataset (shown in Table 2). Instead, Supplementary Figure S6b shows that the thresh-

olding approach increases the representation more than twice for each non-background label class, resulting in an overall increase of five times more non-background pixels in the image patches. By altering the distribution of sampled pixels, pixels can be missed. Supplementary Figure S6c shows the number of missed pixels per class after 80.000 sampled images. It can be seen that while 97% of all unique trace pixels are covered, only 58% of the unique background pixels are covered. Further experiments show that after the full training duration of 400.000 data points, 99.7% of the trace pixels and 80 % of the background pixels are covered. From the increased mIoU shown in Supplementary Figure S6a, we conclude that the benefit of oversampling the trace pixels outweighs the missed background pixels.

In Supplementary Figure S7, we investigate hyperparameters for training trace classifications. These results were obtained after ImageNet prtraining. S7a shows that the mIoU of trace recognition is stable for weight decay values between 0 and 10^{-4}. S7b furthermore shows that mIoU is stable across batch sizes between 5 and 200 when the learning rate is scaled linearly with the batch size.

We downsample the microtrace images obtained with our automated microscope from 1 µm/pixel to 4 µm/pixel (see Subsection IV.D). S7c shows the mIoU we achieve by instead training on the full resolution or training on half the resolution (2 µm/pixel). For each experiment, we train on 400.000 image patches with a field of view (FOV) of 1024 µm. S7d shows the training duration of each experiment. While S7c shows that the impact of removing or weakening the downsampling operation is small (±0.04 IoU), S7d shows that downsampling allows a strong reduction in training time. Compared to training on the full resolution, downsampling to 4 µm/pixel allows a sixteen-fold decrease in training time. Halving the resolution results in a four-fold decrease of training time.

In S7e and f, we report mIoU values obtained when varying the magnification and aspect ratio augmentation bounds during the microtrace training. S7e shows that a magnification augmentation bounded between 0.33× and 3× benefits the mIoU with respect to using no magnification. S7f shows that the mIoU remains stable for training with aspect ratios multiplied up to four times. Zoom and aspect ratio augmentations within these bounds benefit training by reducing the reliance on feature diversity in the training data. Heavier zoom and aspect ratio augmentation are found to deteriorate the training process. This is caused by the induced loss of information by these heavy augmentations.

III. Discussion

In this work, a deep learning model for recognising microtraces in tape lift scans is proposed. This model rapidly processes microtrace scans to provide experts with overviews of traces on tapes. Hereby, it aids experts in their analysis and streamlines trace investigation.

We show that our deep learning model is able to perform semantic segmentation of fibres, hairs, skin, glass and sand with an intersection over union of 0.55 while only being trained on an area of 2.2 dm^2 of annotated tape lift samples. By first pretraining on the large scale ImageNet database of generic everyday images and subsequently pretraining with SSL on unnannotated microtrace images, we outperform a counterpart model trained from scratch with an increase of 0.22 IoU. When we reduce the available annotated data four-fold to 0.6 dm^2, our pretrained model still outperforms training from scratch with all annotated data with 0.02 IoU.

As tape lift scans typically contain a small number of traces scattered across a large background area, 99% of the image area of our dataset represents background (see Table S4). Our proposed image extraction method involving thresholding allows a +0.10 mIoU increase with respect to processing the image area uniformly by effectively oversampling foreground areas.

Combining SSL pretraining with ImageNet pretraining is found to result in the highest mIoU benefits. Solely applying SSL pretraining is found to result in lower mIoU values than ImageNet pretraining, although it provides a strong alternative in situations where ImageNet pretraining is not available. The finding that ImageNet pretraining outperforms sole SSL pretraining is in line with

[15], in which SSL pretraining is investigated for semantic segmentation of biomedical microscopy images, although here another SSL framework (SimCLR [12]) is used. Moreover, it aligns with [76], in which BYOL is compared to ImageNet pretraining for semantic segmentation of natural images [76]. However, [28] and [54, 80], find that BYOL and SimCLR result in higher accuracies than ImageNet pretraining for semantic segmentation of natural images.

We suggest further research into closing the gap between the mIoU benefit of SSL pretraining compared to ImageNet pretraining. Primarily, we see potential in adapting the SSL-framework for segmentation. The BYOL framework, as well as other recent SSL-frameworks such as [13, 14, 30] regard image-level comparison. This bears similarities with image-level classification, the field in which these frameworks show the most notable pretraining benefit [15, 28]. Semantic segmentation on the other hand regards dense (pixel-level) predictions, which motivates the exploration of SSL-frameworks with dense comparison of image pairs [76, 80]. Another approach would be to pre-segment individual traces, extract only the image area containing that trace and subsequently generate an image pair of different views of only that segment. With this approach, an SSL-framework such as [28] can be used for segment-level comparison, possibly benefitting segment-level classification.

Moreover, pre-segmenting traces allows to ensure that traces are contained in both images of the SSL pretraining image pairs. Our current translation augmentation is agnostic of trace locations, allowing a trace to move out of view for one of the images of the pair. This limits SSL pretraining benefit.

Our architecture for semantic segmentation follows [28], where it is found to achieve an mIoU of 74% on the Pascal VOC2012 challenge [24] of semantic segmentation of natural images. In [11], an architecture is proposed that achieves 84% on the same dataset. Further research can be done to determine whether the architecture of [11] also yields an increase in mIoU for the segmentation of microtrace images.

Lastly, we recommend further research in alternative microscopy modes, such as reflection imaging, to improve the classification of opaque, thick traces (see Subsection II.A).

Our study sets a new baseline for forensic microtrace recognition and gives practitioners insight into the benefits of pretraining, the required annotation workload, augmentation parameters and efficient image extraction for microtrace recognition in tape lift scans. We provide trained models and facilitate analysis of model predictions through a graphical user interface [5] to yield a valuable tool for microtrace recognition that aids forensic experts in their investigation. Hereby, we contribute to automating the microtrace finding process and decreasing the human labour required for trace investigations.

IV. Methods

Supplementary Figure S8 and S9 show the architectures of the neural networks. Supplementary Table S1, S2 and S3 respectively summarise the hyperparameters, augmentations and initialisations used in training.

In the following sections, we provide details on our method and substantiate our hyperparameters in the order of Figure 1. We discuss training the pixel classifications in Subsection IV.A, ImageNet pretraining in IV.B, SSL pretraining in IV.C and our image acquisition and extraction in IV.D. Finally, we provide details on our dataset in IV.E and our evaluation protocol in IV.F.

A. Trace training

We train the pixel classifications as shown in Figure 1a. Following [28, 30], we use a fully convolutional network (FCN) architecture [67] with a ResNet-50 feature extractor [32]. The last block of the feature extractor (*stage 5*) uses dilated convolutions to increase the resolution of the feature map and thereby benefit its suitability for dense prediction tasks such as semantic segmentation [10]. The full architecture and output dimensions of each block are shown in Supplementary Figure S8.

Following [67], the transpose convolutional layers are initialised as bilinear interpolation upsamplers.

The classifier layers are initialised randomly (He uniform [31]).

The network is trained by minimising the difference between the predicted pixel classifications and the expert annotated classifications via standard per-pixel softmax cross-entropy loss [53, 62]. We do not freeze the weights of pretrained feature extractors but instead allow all network weights to be optimised.

We use a batch size of 160, a weight decay of 0.0001 and a base learning rate of 0.016, which is multiplied by 0.1 at the 70th and 90th percentile of training, following [28] for semantic segmentation. Here, we use an Stochastic Gradient Descent (SGD) optimiser with a momentum factor of 0.9. We use 2 million data points across the annotated tape lift area for training from scratch. For all other models, we train on 400.000 points due to the earlier convergence. This can be seen in Supplementary Figure S10.

B. ImageNet pretraining

For ImageNet pretraining (illustrated in Figure 1b), we use the pretrained ResNet-50 weights provided by [53]. This model was optimised for 90 epochs on the 1.3 million images of ImageNet [64], resulting in 115 million data point steps. Here, a batch size $m = 32$ was used in combination with a learning rate $\eta = 0.1$ that was multiplied by 0.1 after 30 and 60 epochs.

C. Self-supervised pretraining

In Figure 1c, a feature extractor is pretrained with SSL on image patches extracted from unannotated microtrace scans. We employ the SSL model 'Bootstrap Your Own Latent' (BYOL) [28]. With BYOL, the feature extractor is rewarded for predicting the similarity of the feature representations of two augmented image crops of the same underlying structure. To generate these augmentations, we extract pairs of image patches with the method described in Subsection IV.D. For each coordinate, we extract two images with different random augmentations. Here, we also introduce a translation augmentation with a maximum distance d_m to push the image patches apart. The translation augmentation and zoom augmentation determine the image crop within the microtrace scan. As the cropping operation is the essential element to learning useful representations [28], we performed a grid search optimisation on d_m for the translation augmentation and the magnification bounds of the zoom augmentation (see Supplementary Figure S4).

BYOL is composed of an asymmetric architecture of two separate networks: an online network and a target network, as shown in Figure 1b. The online network is composed of a ResNet-50 feature extractor, an multilayer perceptron (MLP) projector and an MLP predictor. The target network is composed of a second feature extractor and a second projector, both with different weights than the online network. In Supplementary Figure S9, we provide a detailed overview of the implementation and the output dimensions of each network.

The online network and target network both take one of the augmented patches as input. The weights of the online network are optimised to predict the output of the target network. By introducing asymmetry and only optimising the online network, BYOL prevents the feature extractor from learning a collapsed solution [74]. In a collapsed solution, the model outputs a constant feature representation independent of the input image. This constant representation results in a perfect similarity for each image pair, while it is not useful for classification. Although the weights of the target network feature extractor and projector can be identical to the corresponding weights of the online network [14], BYOL updates the target network with an exponential moving average of the weights of the online network. Hereby, it provides a smoother target representation to optimise the online network on. This results in improved performance with respect to using identical weights [28].

To train the online network, the loss function $\mathcal{L}$ is calculated for pair of augmented images (V_i, V_i') as

$$\mathcal{L} := \left\| \frac{z_\theta(V_i)}{\|z_\theta(V_i)\|_2} - \frac{f_\xi(V_i')}{\|f_\xi(V_i')\|_2} \right\|_2^2 + \left\| \frac{z_\theta(V_i')}{\|z_\theta(V_i')\|_2} - \frac{f_\xi(V_i)}{\|f_\xi(V_i)\|_2} \right\|_2^2. \quad (1)$$

Here, the output of the online network $z_\theta(\cdots)$ is normalised and compared to the normalised output of the target network $f_\xi(\cdots)$. The sum of the mean squared error is taken for feeding V_i to the online network and V_i' to the target network and the other way around.

At each optimisation step k, the online network is optimised to minimise $\mathcal{L}$ and the target network receives an exponential moving average of the parameters of the online network θ (predictor excluded):

$$\theta_{k+1} = \texttt{optimiser}(\theta_k, \nabla_\theta \mathcal{L}, \eta) \quad (2)$$

$$\xi_{k+1} = \tau_k\, \xi_k + (1-\tau_k)\, \theta_{k+1}. \quad (3)$$

We use a batch size $m = 80$ and thus process 80 image pairs per optimisation step. Following [28] for small batch sizes, the base learning rate is calculated as $\eta_{\text{base}} = 0.4 \cdot m/256$. The learning rate is scheduled to increase linearly (*warm-up*) from 0 to η_{base} within 1250 steps (100.000 image pairs) and to decay to 0 at the maximum number of steps K with cosine decay [44].

The weight decay is set to $1.5 \cdot 10^{-6}$ independent of the batch size m, following [28].

As proposed in [28], we schedule the exponential moving average parameter τ_k to increase to reach 1 at the maximum number of steps K with

$$\tau_k = 1 - (1-\tau_0)\ \frac{\cos(\pi k/K) + 1}{2}. \quad (4)$$

In [28], $\tau_0 = 0.996$ and $\tau_0 = 0.9995$ are proposed for batch sizes $m = 4096$ and $m = 512$ respectively. We further scale τ_0 from a reference batch size m_{ref} to a target batch size m_{tgt} with the exponential moving average parameter scaling rule proposed in [81]:

$$\tau_{0,\text{tgt}} = \tau_{0,\text{ref}}^{m_{\text{tgt}}/m_{\text{ref}}}. \quad (5)$$

Filling in $\tau_{0,\text{ref}} = 0.996$, $m_{\text{ref}} = 4096$ gives $\tau_{0,\text{tgt}} = 0.9995$ for $m_{\text{ref}} = 512$, matching τ_0 proposed in [28]. For our batch size $m = 80$, the scaling yields $\tau_0 = 0.99992$, which is the value we use in our experiments.

We pretrain with SGD [53] using a momentum factor of 0.9 on 10 million image pairs ($K = 125.000$) across the unannotated dataset. This costs four days of training on our hardware. In Supplementary Figure S11, we show test mIoU results obtained for training with fewer image pairs.

D. Image Extraction

We train on microtrace image patches during trace training and SSL pretraining. We extract and augment these patches from transmission microscopy scans of tape lift samples as shown in Figure 1d.

Tape lifts are composed of a variety of small traces scattered on a large transparent sample, resulting in a large part of tape-lift scans to represent background. In our dataset, experts labelled more than 98% of all pixels as background (see Supplementary Table S4). Random sampling image patches from the microtrace samples [16] will therefore result in many patches containing only background. Motivated by the hypothesis that learning trace classifications is harder than learning to recognise background, we aim to oversample foreground regions by employing a thresholding method for extracting training images. While other methods extract patches uniformly and filter out background-rich patches with thresholding [15, 85], we apply thresholding earlier in the sampling process and extract only patches centred around foreground areas.

Transmission microscopy scans show dark objects across a contrasting white background. Moreover, the background is homogeneous. Therefore, thresholding with a single global threshold suffices to distinguish foreground areas from background [78]. To choose this threshold, we employ the histogram-based triangle method [82], which is suited for images dominantly consisting of background [40, 63, 78]. This is shown in Supplementary Figure S12. With the found threshold, a total of 94% of the image area in the annotated dataset is estimated as background. The remaining 6% is estimated as foreground, including possible microtraces or artefacts.

Training coordinates are then sampled randomly (uniform) from the thresholded foreground areas. At each sampled coordinate, an image patch of size 256×256 pixels is extracted in a resolution of 4 µm per pixel. We augment the image patches to artificially enlarge the dataset. Specifically, we make a crop of random zoom, aspect ratio and rotation and resize it to 256×256 pixels as shown in Supplementary Figure S13. Then, we further augment the image patches by recolouring, blurring and mirroring, which results in the required training images. The chosen augmentations, randomisation processes and their parameters are derived from [28] and are listed in Supplementary Table S2.

For trace training, we extract single image patches with corresponding annotation masks. For SSL pretraining, we extract pairs of similar images. For evaluating the model, we use no thresholding and sample image patches uniformly across the test set as described in the next section.

We sample patches with a FOV of 1024×1024 µm per patch to ensure that the majority of the traces can be captured fully within the FOV (see Table 2). We mean-downsample the images to a resolution of 4 µm/pixel, thus resulting in a patch size of 256×256 pixels.

E. Dataset Acquisition

The samples were acquired by tape-lifting cotton sheets, faces and hands. Furthermore, tape samples were compiled with glass and sand particles and dandruff of volunteers.

The samples were scanned using transmission microscopy with the automated microscope developed in [70]. This device illuminates the sample from below and captures neighbouring images from the opposite side in a resolution of 1 µm/pixel. After applying a shading correction filter, the images are stitched together to a full-size `.tiff` file (6.4 Gigapixel for a tape of 80×80 mm). In total, we scanned 315 tape samples of various sizes and annotated 10 of them (see Table 1).

The tapes were annotated with open-source software [5]. To minimise the annotation workload, we first generated pseudo-annotation masks based on simple heuristics and manually corrected these. Specifically, we first segmented traces with thresholding and classified each region based on the histogram and a set of shape features. We estimate the time spent on correcting and manually annotating trace images on 120 hours for our annotated dataset composed of approximately 21.000 image patches (see Table 1). Note that the large background ratio accelerated the annotation process. For pixel-wise annotations on everyday photographs, composing a dataset of this size is estimated to require more than 500 hours [52].

In Table 2, we provide the occurrence of each trace label in the annotated training dataset. To investigate the class balance in the unannotated dataset, we perform inference with the model presented in Subsection II.A on the unannotated tapes. We report the results in Supplementary Table S4 and show a random selection of the predictions in Supplementary Figure S14. According to the model, the relative occurrence of glass and sand in the unannotated dataset is approximately seven times smaller than their occurrence in the annotated dataset. Compared to the annotated dataset, the unannotated dataset mainly considers fibres and skin.

We split our annotated dataset in an approximate 3:1 train:test ratio in terms of tape lift area. In particular, we use an 80×80 mm tape lift containing all trace types as test set and use the other nine annotated tapes of various sizes for training as shown in Table 1.

F. Evaluation

After training trace classifications, we test the model on an annotated microtrace scan excluded from the annotated training dataset and the unannotated dataset that is used in SSL pretraining. This scan consists of an 80×80 mm tape lift sample containing all trace types, shown in Supplementary Figure S15. During testing, the scan is processed via a uniform grid of image patches (see Supplementary Figure S5), without oversampling the foreground with thresholding or applying augmentations as described in Subsection IV.D. For each image patch, the predictions at the edges are discarded, following [46, 62]. These predictions are of lower accuracy due to the lack of contextual

information. Therefore, the FOV of each image patch is increased during testing with 12.5%, resulting in partially overlapping input patches.

The trace recognition performance of the model is evaluated with the mean Intersection over Union (mIoU) [16, 23]. For each class A, the IoU is defined as the number of correctly identified trace pixels divided by the sum of all pixels either predicted or annotated as class A:

$$\mathrm{IoU}_A = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP} + \mathrm{FN}}, \tag{6}$$

Here, TP denotes the number of true positives, while FP and FN denote the number of false positives and false negatives, respectively. The maximum value of 1 indicates that the annotations and predictions match perfectly. Conversely, the minimum score of 0 indicates no true positives were predicted. For overall performance, the mean of the IoU (mIoU) of the n_c non-background classes is taken:

$$\mathrm{mIoU} = \frac{1}{n_c} \sum_{A=1}^{n_c} \mathrm{IoU}_A \tag{7}$$

In our case, $n_c = 5$, with the classes being fibre, glass, hair, sand, and skin. Similar to the evaluation protocol of [8], we accumulate the confusion statistics over the test set as single image patches typically do not contain all of the five trace classes. We calculate the overall mIoU based on the accumulated statistics over the entire test set. The violin plots used throughout the report show the distribution of mIoU values that are obtained by accumulating over a randomly determined 10-fold split in the test set that was kept constant throughout the experiments.

To investigate label-efficient learning, we split our annotated training set into equally sized regions as shown in Supplementary Figure S16 and vary the number of regions used in training the trace classifications.

V. Acknowledgements

We thank Hellenic Police and Staffordshire University for sharing scan records of tape lifts and thereby contributing to the dataset. Furthermore, we gratefully acknowledge the financial support of the Netherlands Forensic Institute (NFI). Lastly, we acknowledge Spectricon for the support regarding the SMMART automated microscope.

Tables

Table 1: Overview of datasets. The last column shows the number of unique image patches that can be extracted for a field of view (FOV) of 1024×1024 µm. The unannotated dataset includes the tapes of the annotated training dataset. Thus, a total of 315 samples were scanned.

Dataset	#tapes	Total area	#patches
Unannotated	314	1.082 m^2	$1.0 \cdot 10^6$
Annotated			
Training	9	0.022 m^2	$21 \cdot 10^3$
Test	1	0.006 m^2	$5.8 \cdot 10^3$

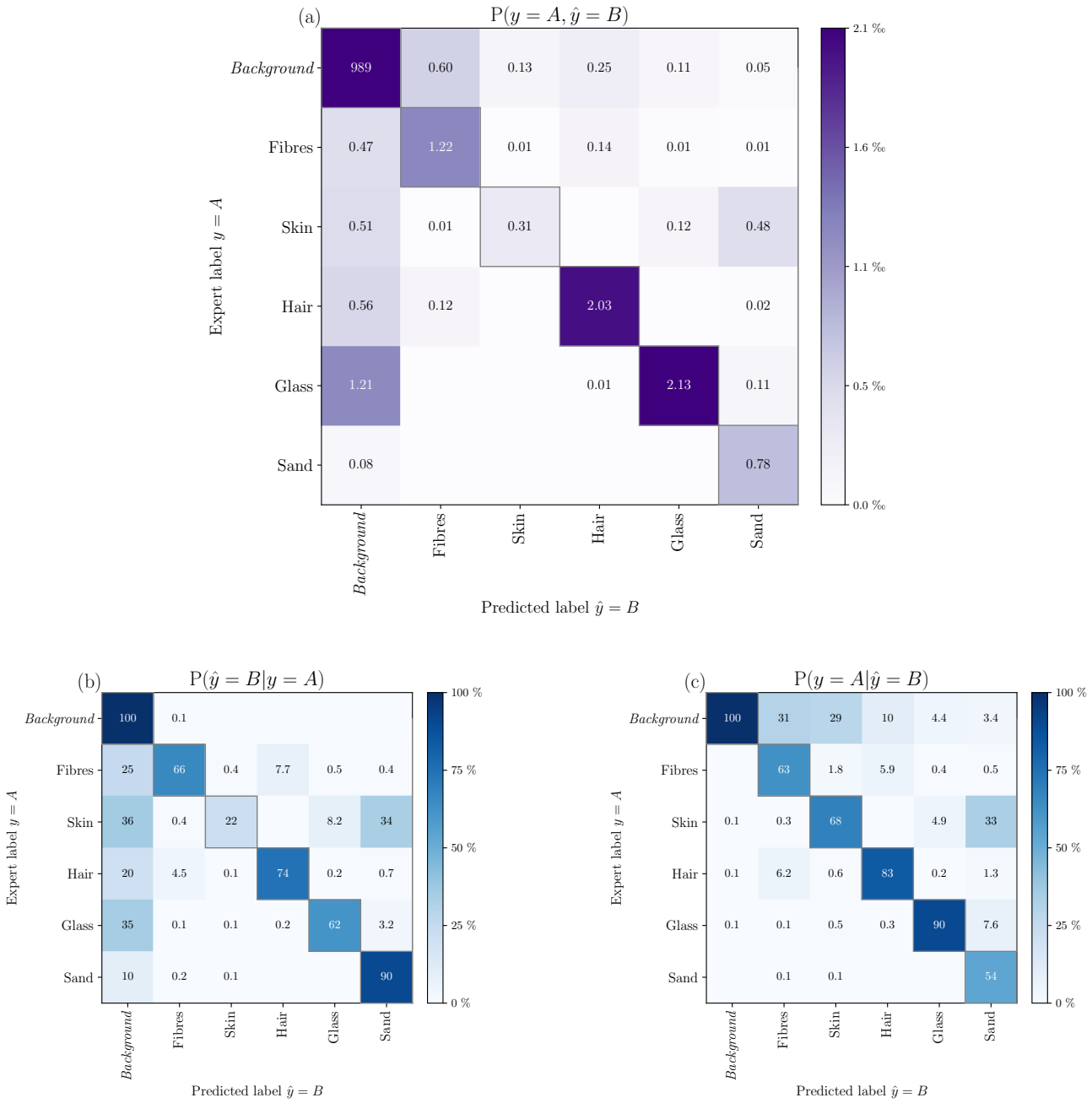
Table 2: Traces in annotated training dataset. The class skin is segmented in regions instead of individual cells, therefore yielding the two right columns ill-defined. The trace size is calculated as the median diameter of the minimum bounding circle for each annotation. For the uncertainty measure, the median absolute deviation is used.

Class	Total area	Trace size	Count
Fibres	117 μm^2	0.5 ± 0.2 mm	$3 \cdot 10^3$
Hairs	47 μm^2	7 ± 4 mm	$2 \cdot 10^1$
Glass	47 μm^2	0.2 ± 0.1 mm	$2 \cdot 10^3$
Sand	20 μm^2	0.3 ± 0.1 mm	$4 \cdot 10^3$
Skin	79 μm^2		

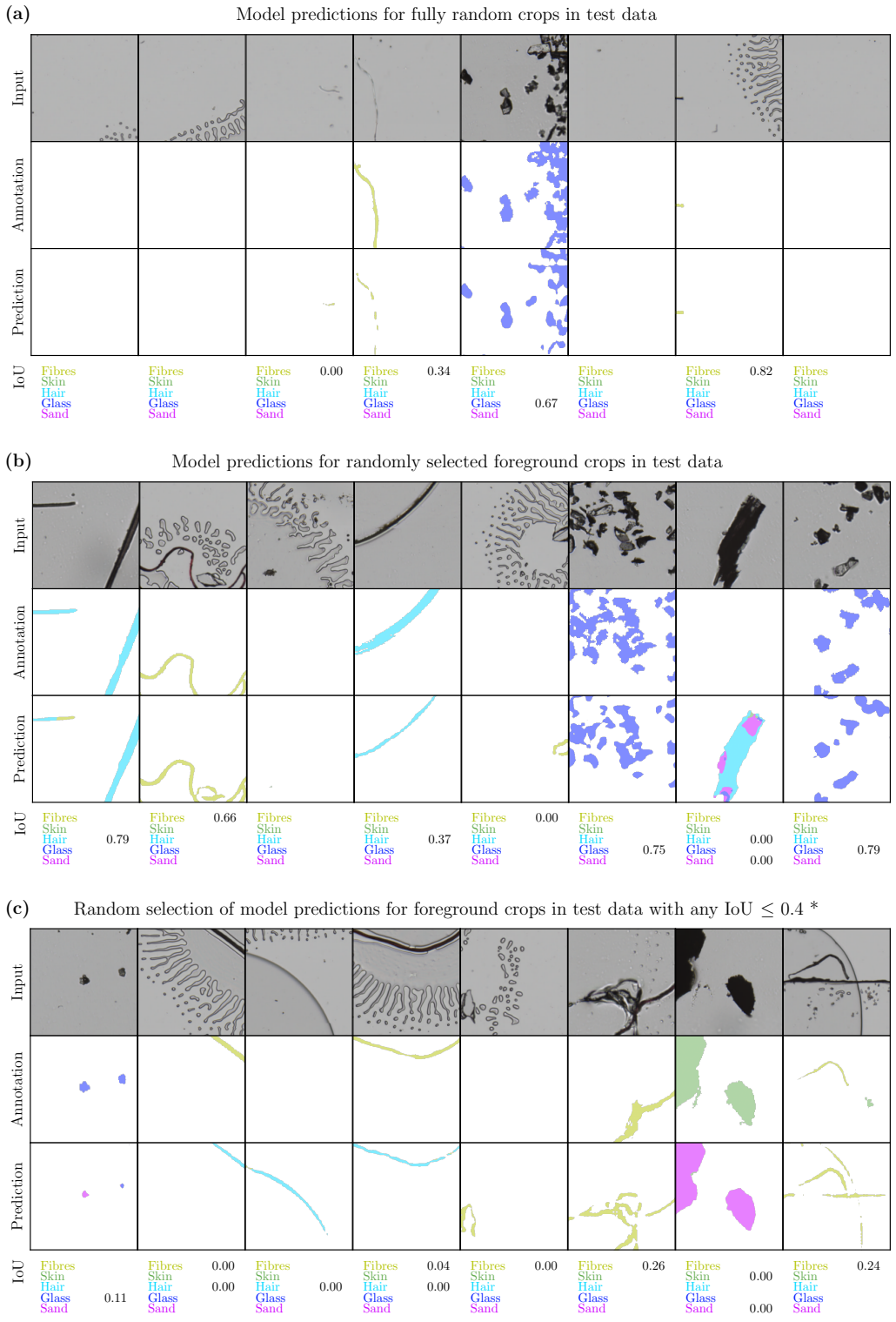
Supplementary materials

List of Supplementary Figures

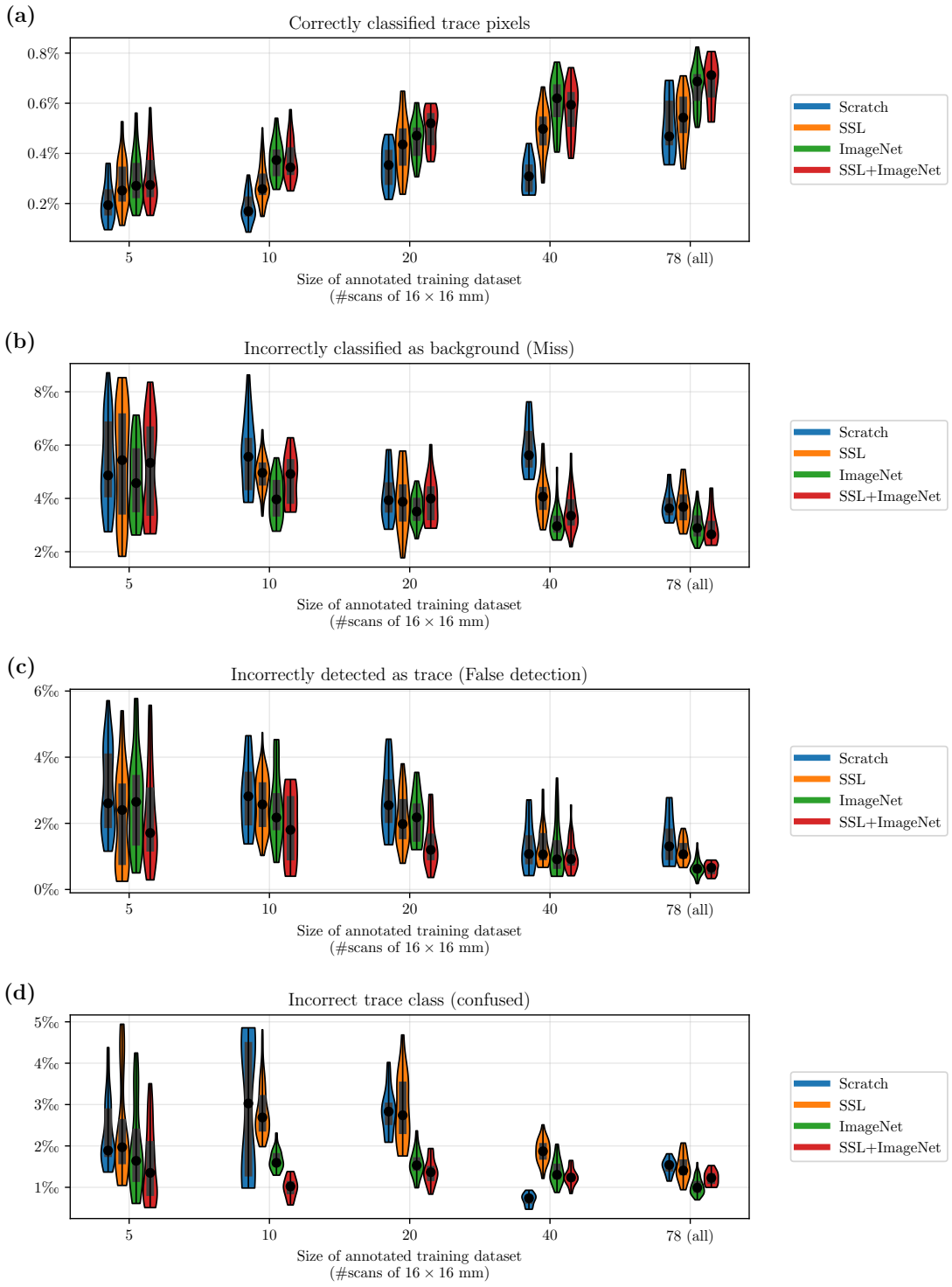
List of Supplementary Tables



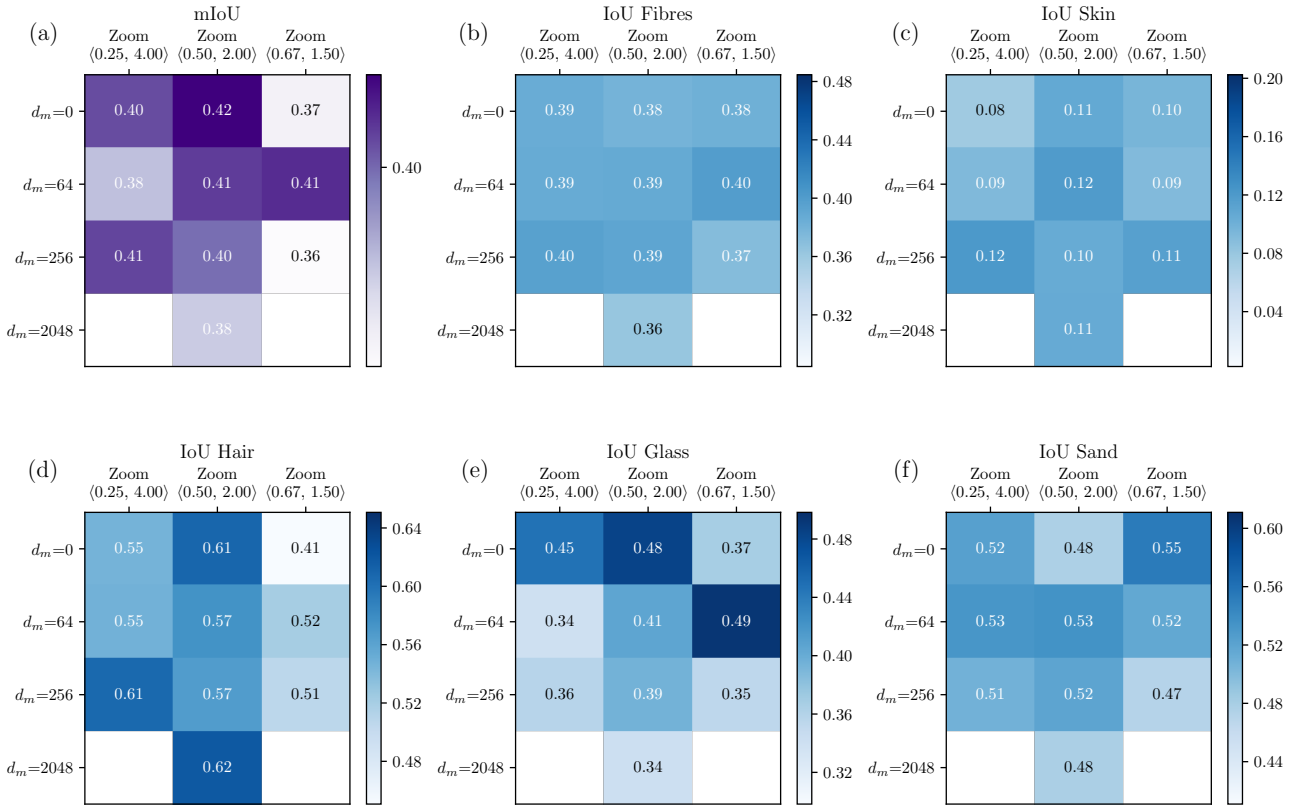
Supplementary Figure S1: Confusion matrix of predictions in test set. (a) Normalised with total number of predictions. Values lower than 0.005 ‰ are omitted. (b) Row-wise normalised, yielding probabilities conditioned with true values. The diagonal presents the recall (sensitivity) per trace class. Values lower than 0.05% are omitted. (c) Column-wise normalised, yielding probabilities conditioned with predicted values. The diagonal presents the precision per trace class. Values lower than 0.05% are omitted.



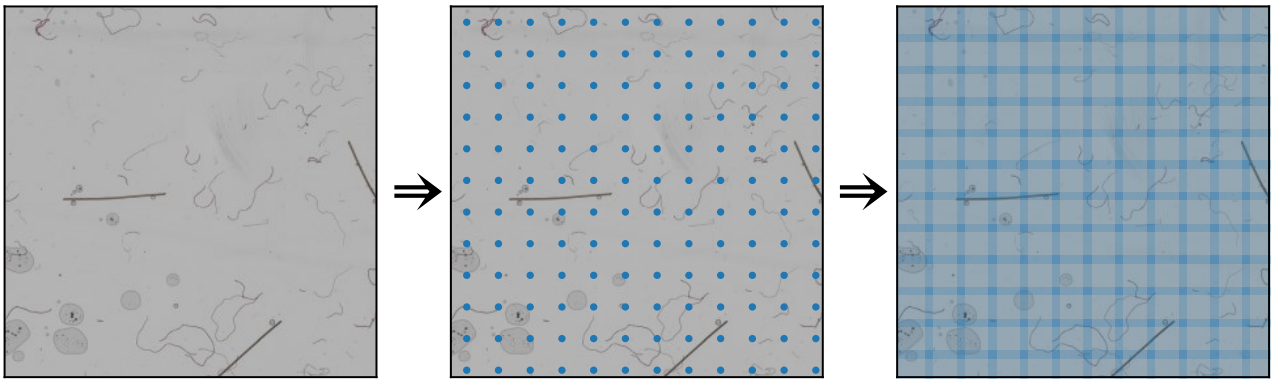
Supplementary Figure S2: Selection of model predictions in test dataset to visually assess model performance and corresponding IoU values. The IoU values for classes that span less than 1% of the total image area in both the expert annotations and the predictions are omitted. (a) Predictions for fully random crops in test data. Here, the location of the crops was selected via uniform sampling. (b) Predictions for randomly selected foreground crops. These crops were extracted with the thresholding approach discussed in Subsection IV.D. (c) Foreground crops where any of the IoUs for the classes that span at least 1% of the image area in either the predictions of the annotations, is lower than 0.4.



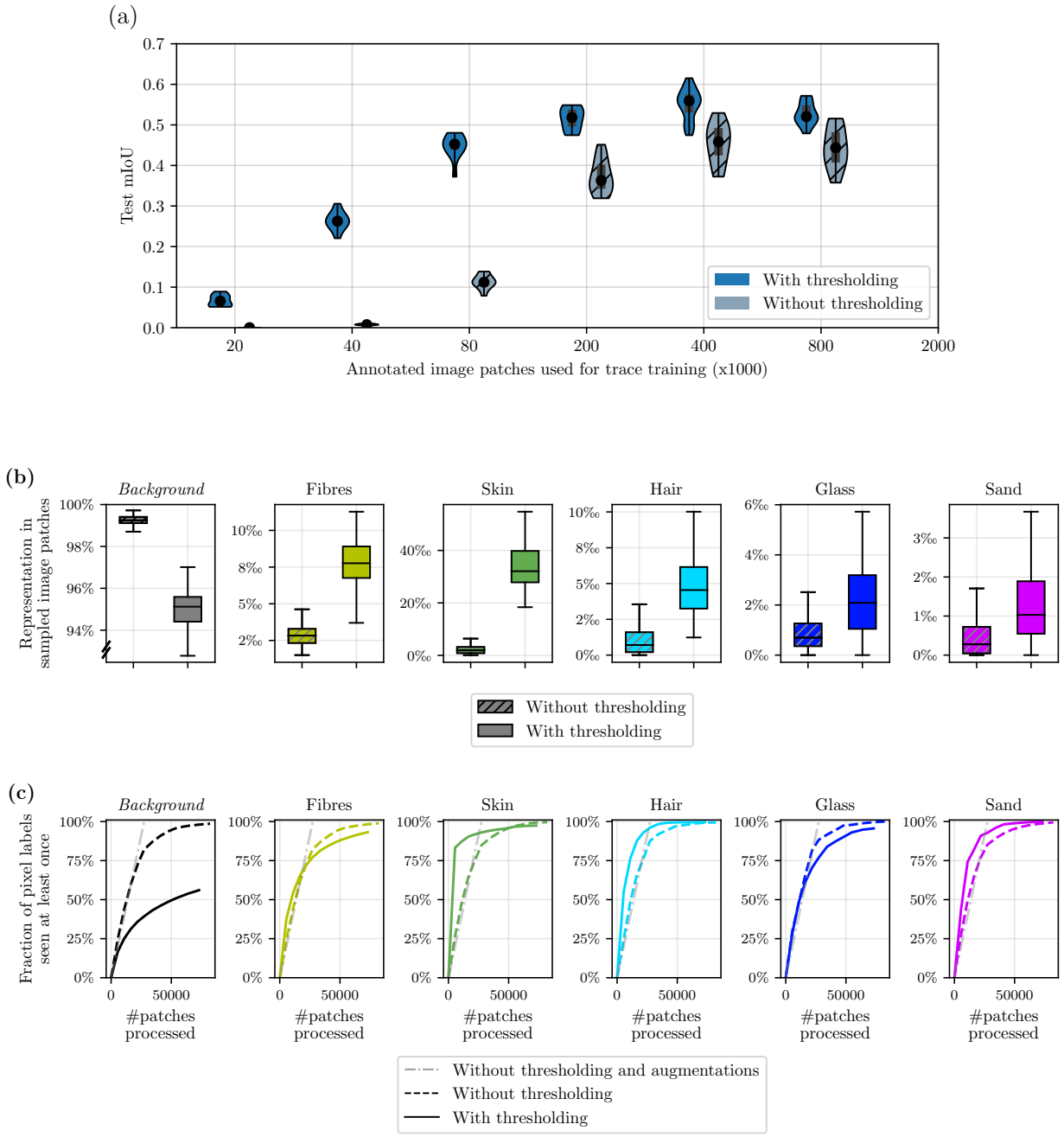
Supplementary Figure S3: Analysis of error modes for label efficient learning. Note that the y-axis presents promille values. The remaining $\sim 99\%$ of predictions consider correctly classified background.



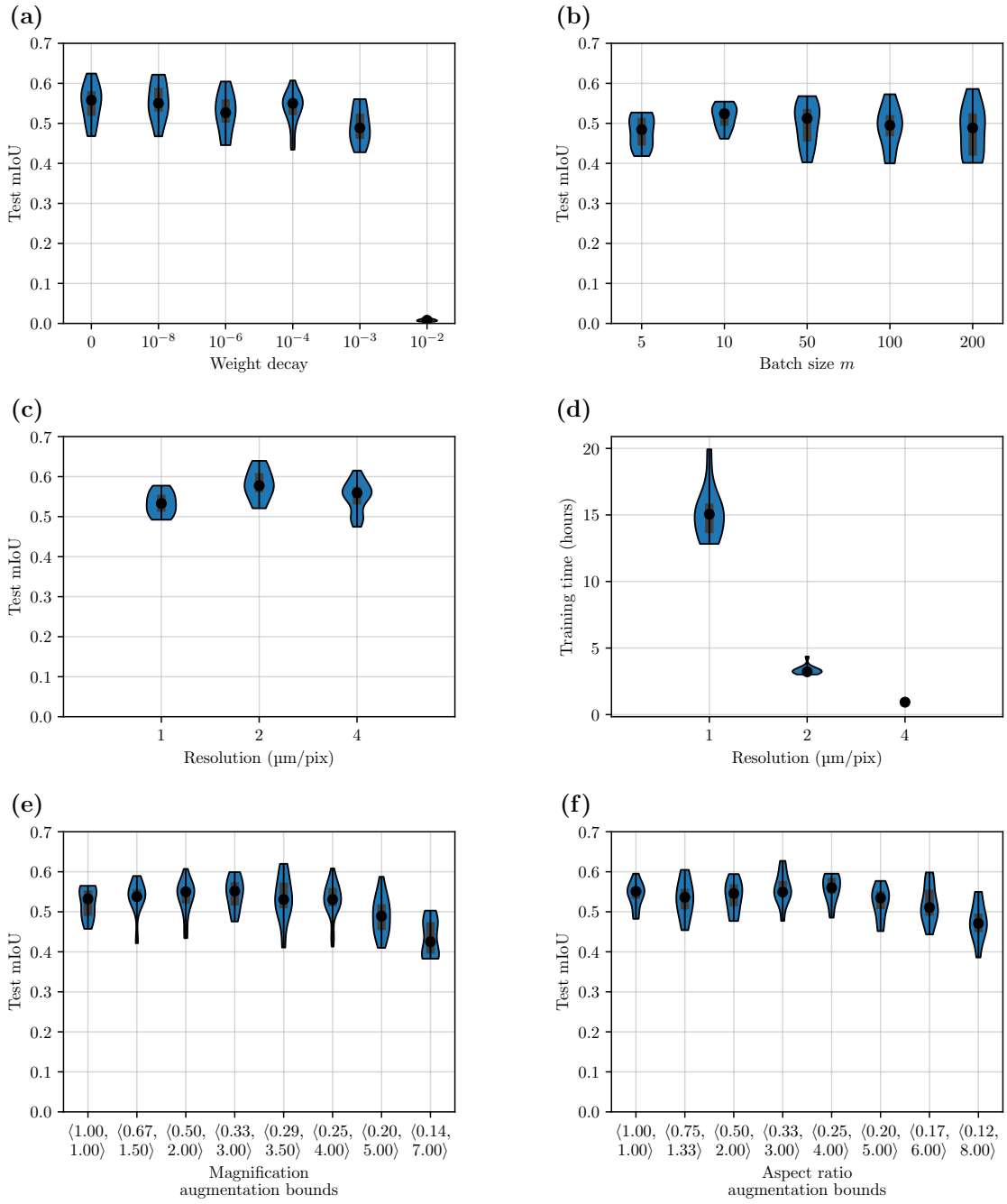
Supplementary Figure S4: Test mIoU after pretraining with different cropping parameters. The horizontal axis shows the zoom, the vertical axis the maximum translation. The other augmentation parameters are kept constant and are listed in Supplementary Table S2. To reduce training time, we train with 7 million data points for this experiment instead of 40 million (see Supplementary Figure S11). (a) shows the overall test mIoU, (b-f) show the IoU per class.



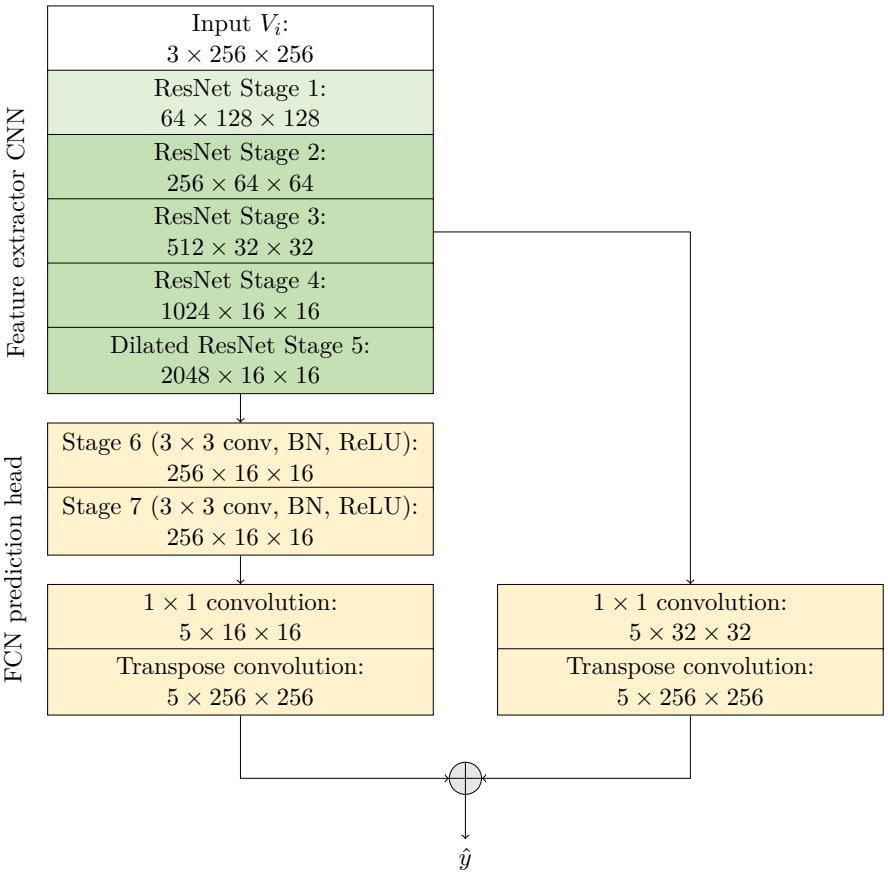
Supplementary Figure S5: Image extraction without thresholding. This approach is used for evaluating the trace recognition models on the test set and for the experiments of Supplementary Figure S6. Opposed to extracting images as in Figure 1d, the image patches are now uniformly distributed across the scan. Again, the squares and dots visualise the extracted patches and their centre points respectively. The patches contain a 12.5% overlap in FOV (see Subsection IV.F).



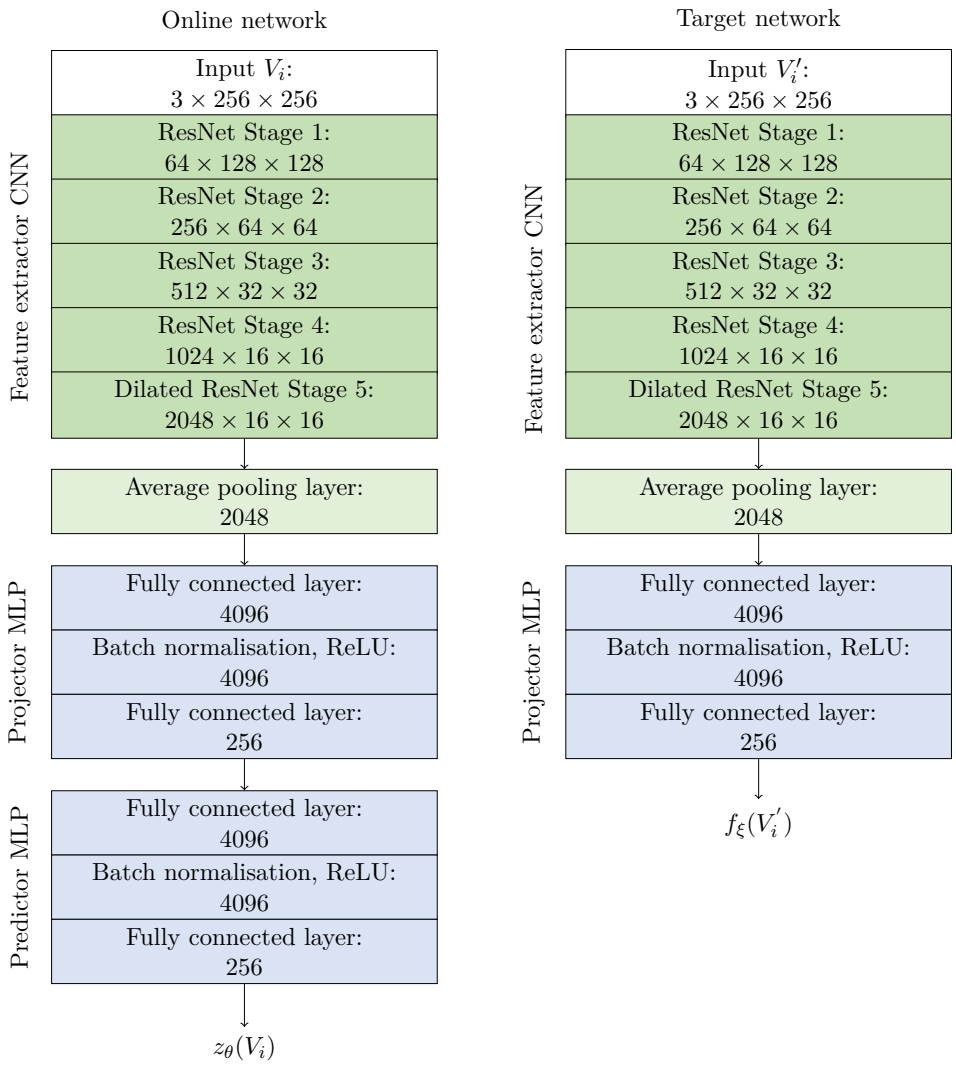
Supplementary Figure S6: Benefit of thresholding approach shown in Figure 1d compared to the uniform approach shown in Supplementary Figure S5. For the uniform approach, processing 21.000 image patches corresponds to one data set iteration (see Table 1). This is called one *epoch*. (a) mIoU of trace recognition for training with and without thresholding approach. The test mIoU is shown for training with various amounts of image patches across the annotated dataset. Here, ImageNet pretraining was used. It can be seen that the thresholding approach converges earlier and to a higher final accuracy. (b) Efficacy of thresholding approach to oversample the foreground pixels. The class distribution of pixels over 8000 randomly selected image patches is shown. The left boxplots show the distribution of patches sampled without thresholding, the right boxplots show the class distribution for sampling with thresholding. It can be seen that the thresholding approach results in oversampling the non-background classes. (c) Analysis of missed pixels in the annotated dataset due to thresholding. It can be seen that after processing 80.000 image patches (4 epochs for the uniform approach), 95% of the non-background pixels have been seen at least once. Further experiments show that after 400.000 image patches (20 epochs for the uniform approach), 99.7% of all non-background pixels are seen, while 80% of the unique background pixels are seen.



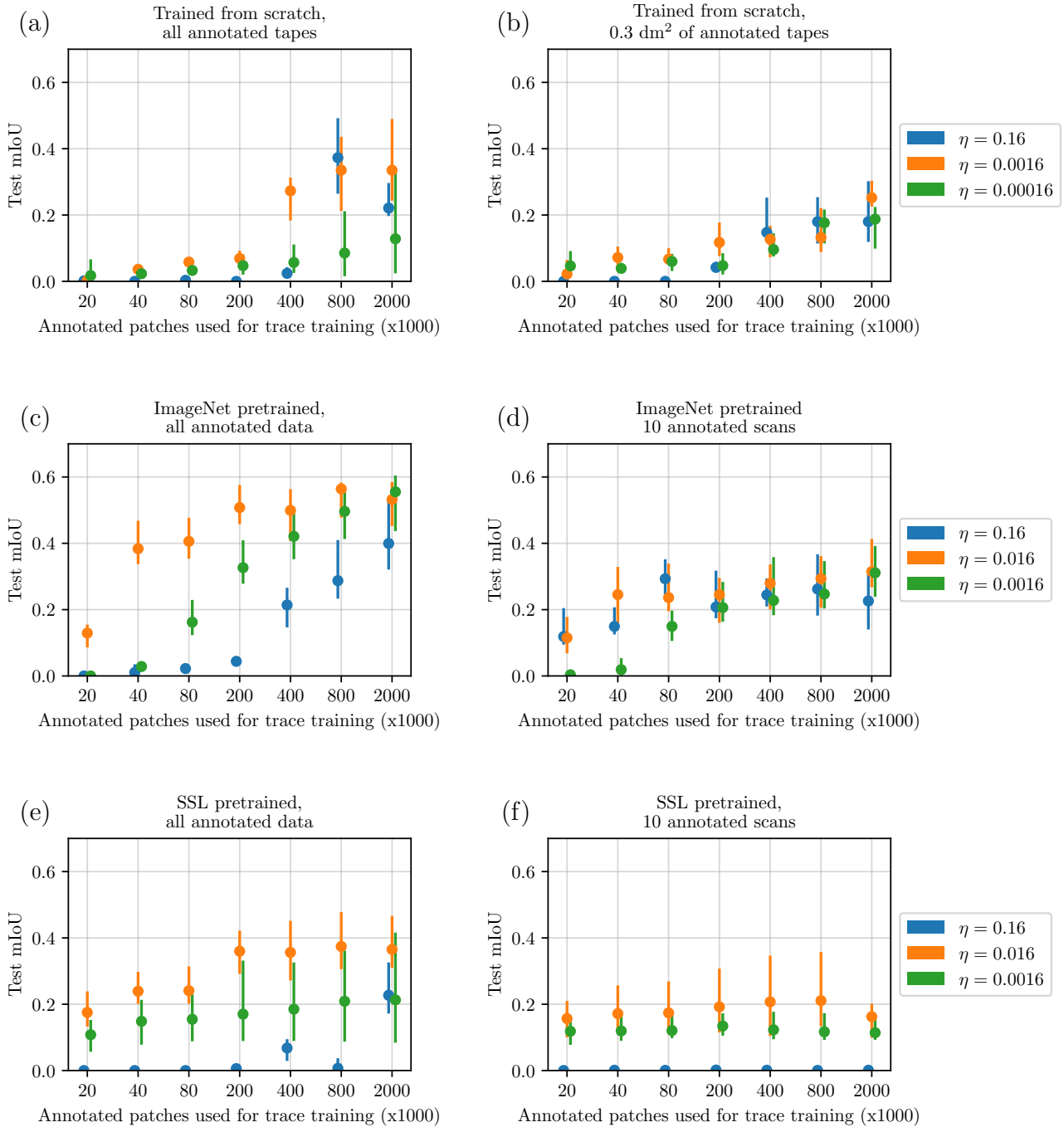
Supplementary Figure S7: Test mIoU for various hyperparameters for microtrace training on all annotated images after ImageNet pretraining. Unless stated otherwise, each experiment is trained with batch size $m = 160$, weight decay 10^{-4}, base learning rate $\eta = 0.001 \cdot m$, 400k image patches, a resolution of 4 µm/pix, a field of view of 1024 µm, magnification bounds $\langle 0.5, 2 \rangle$ and aspect ratio bounds $\langle 3/4, 4/3 \rangle$. (a) Weight decay. (b) Batch size. (c) Image resolution. The annotation masks have a fixed resolution of 4 µm/pix. The upsampling factor of the transpose convolutional layers was adjusted accordingly. m was set to 10, 40 and 160 from left to right. (d) Training time required to process 400.000 image patches.(e) Bounds of zoom augmentation. (f) Bounds of aspect ratio augmentation.



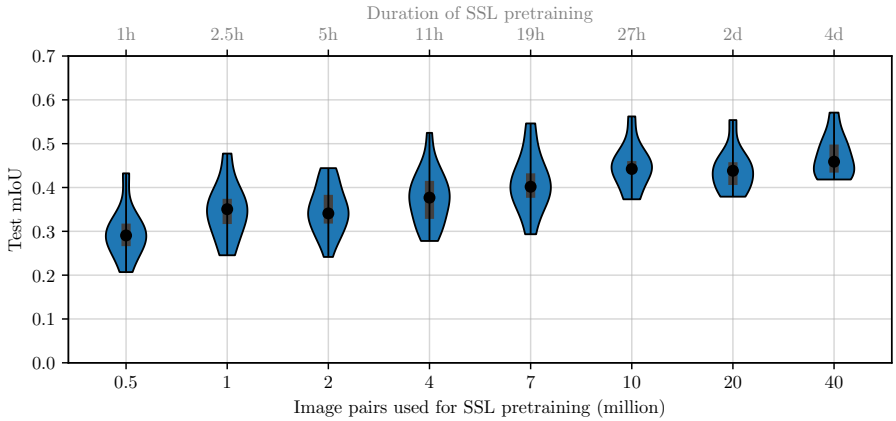
Supplementary Figure S8: Architecture for pixel-wise classification of microtraces with output sizes per block for an image 256×256 pixels. The architecture is similar to the semantic segmentation architecture used in [28, 30], using a ResNet-50 backbone [32] with dilated convolutions in stage 5 in combination with a fully convolutional network (FCN) [67] prediction head.
Two convolutional blocks are inserted after the backbone. These blocks each consist of a 3×3 convolution with 256 channels, followed by batch normalization and ReLU activation. After these two blocks, a 1×1 convolution transforms the 256 channel feature vectors to classifications. These predictions are 16 times upsampled via a transpose convolutional layer with kernel size 8 and stride 4.
Furthermore, a skip connection takes the output of the stage 3 block to generate predictions based on the 32×32 feature map via a 1×1 convolutional layer. These predictions are 8 times via a transpose convolutional layer with kernel size 4 and stride 2 and summed together with the predictions that were made on the 16×16 feature map. Removing the skip connection was found to result in a decrease of 0.075 ± 0.03 IoU for the segmentation of fibres while having a similar performance for other classes.



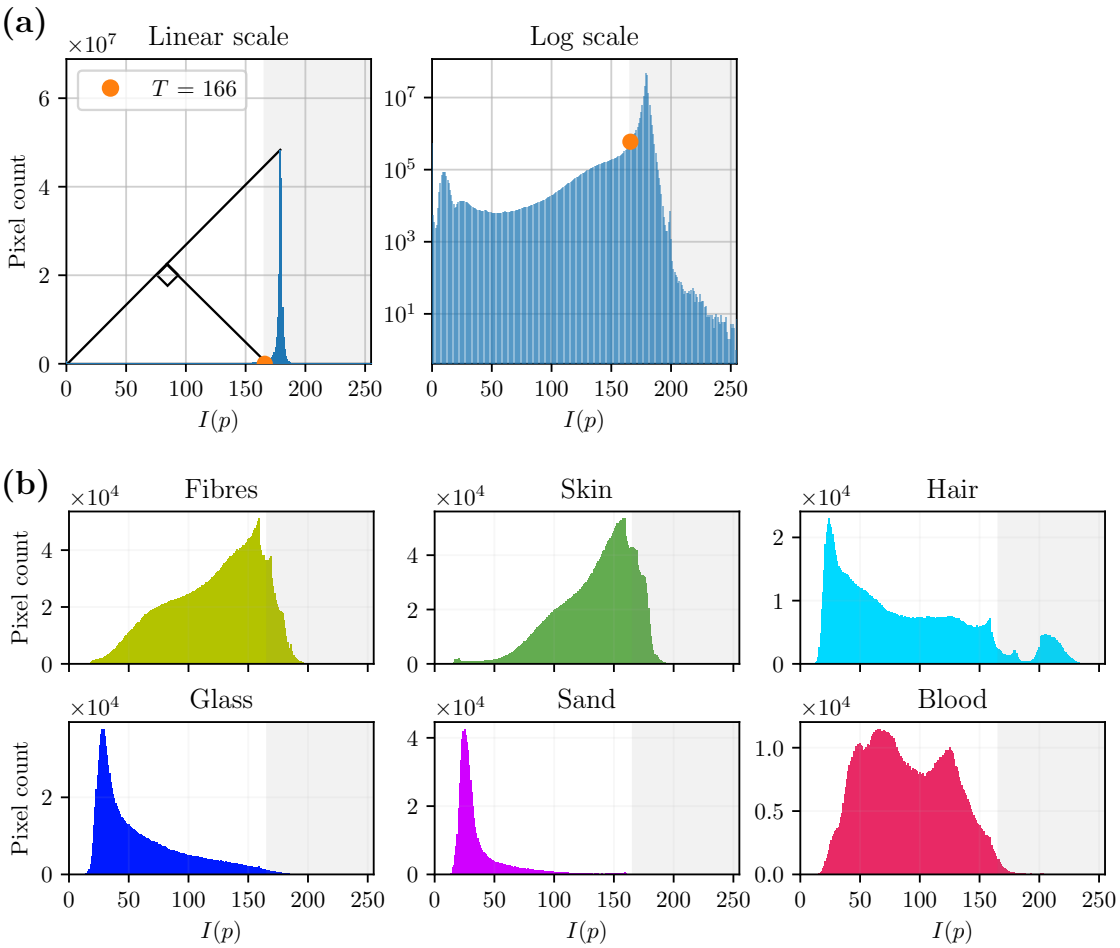
Supplementary Figure S9: Self-supervised pretraining architecture with output sizes per block for an image of 256×256 pixels. The architecture follows BYOL [28]. The feature extractors are composed of the convolutional layers of a ResNet-50-v1 [32], where the 3×3 convolutions of stage 6 use dilation 2 and stride 1. The projector and predictor are MLPs consisting of a hidden layer of size 4096 followed by batch normalisation, ReLU activation and a final layer of output size 256.



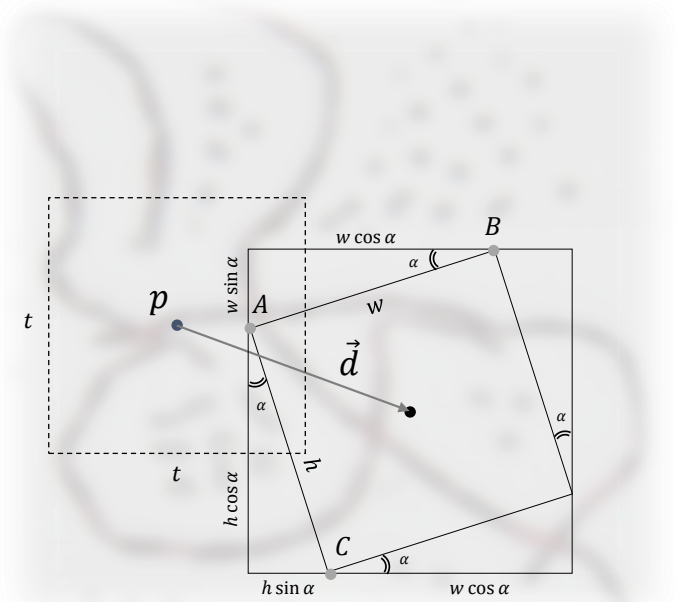
Supplementary Figure S10: Convergence of microtrace training for various learning rates and training lengths. No learning rate scheduling is used in these experiments. A randomly selected single set of 0.3 dm^2 of tapes was selected for the plots in the right column. The overall test mIoU is shown with dots, the vertical lines present the minimum and maximum mIoU over a 10-fold in the test set. It can be seen that choosing $\eta = 0.016$ is a suitable learning rate for each of the experiments. Furthermore, it can be seen that the models trained from scratch take longer to converge. These models take approximately 2 million data points (2.5h on our hardware), while the SSL and ImageNet pretrained models converge in approximately 400.000 data points (30 minutes on our hardware). The convergence of models with combined SSL+ImageNet pretraining is assumed to be similar to (c-f), thus converging in 400.000 data points for a learning rate of $\eta = 0.16$.



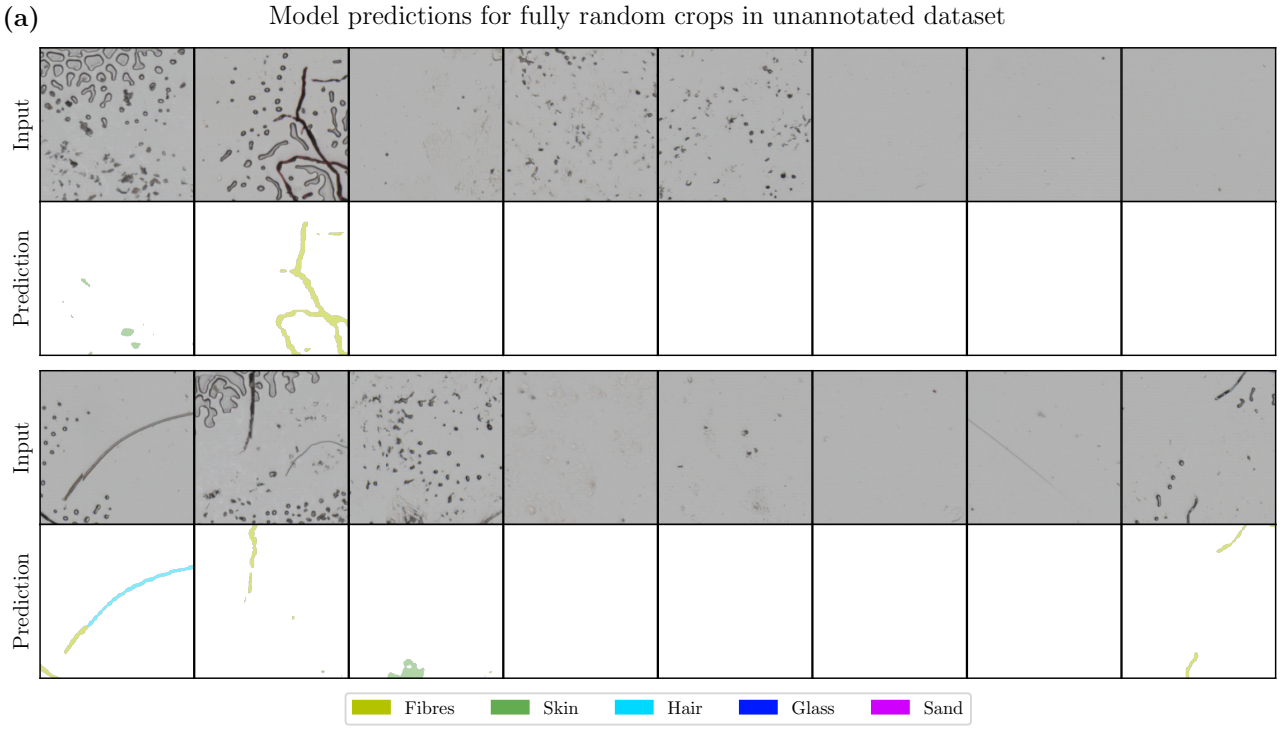
Supplementary Figure S11: Trace recognition mIoU with respect to the number of data points used in SSL pretraining. Each violin presents a separate experiment where the learning rate η and moving average parameter τ are scheduled as described in IV.C with the corresponding number of optimisation steps K. Here, K is calculated as the number of image pairs used in pretraining divided by the batch size $m = 80$. On the top axis, the approximate duration of training on our hardware is given per experiment.

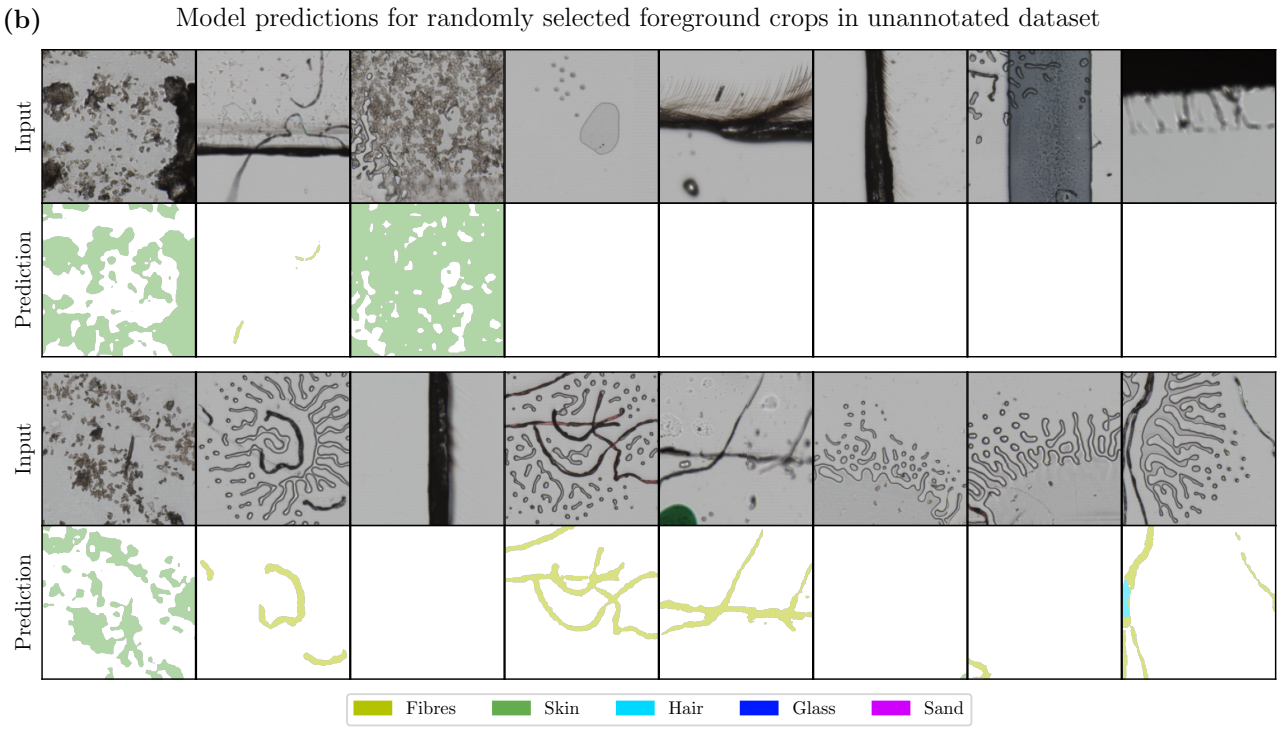


Supplementary Figure S12: Thresholding the pixel intensity to segment foreground. The microtrace scans are mean-downsampled to a resolution of 32 µm/pixel and the grayscale value per pixel $I(p)$ is taken. (a) A threshold is determined with the triangle method [82] based on the histogram of $I(p)$ over 100 randomly selected scans from the unannotated dataset. A line is drawn from the index 0 to the peak. Then, the threshold T is determined by maximising the distance between the line and the histogram, resulting in a threshold of $T = 166$. With this threshold, 94% of the image area of the selected scans is estimated as background: $P_{\text{bg}} = \{p \mid I(p) > T\}$. (b) Analysis of the found threshold with the pixel labels in our annotated dataset. Histograms of $I(p)$ per pixel classification are shown, excluding the test data. The grey-marked area exceeds the threshold of $T = 166$ and is assumed to represent background area with the thresholding operation. It can be seen that although most trace pixels are segmented as foreground correctly, a part of the trace pixels is incorrectly labelled as background. The effect of this is discussed in Subsection II.D.

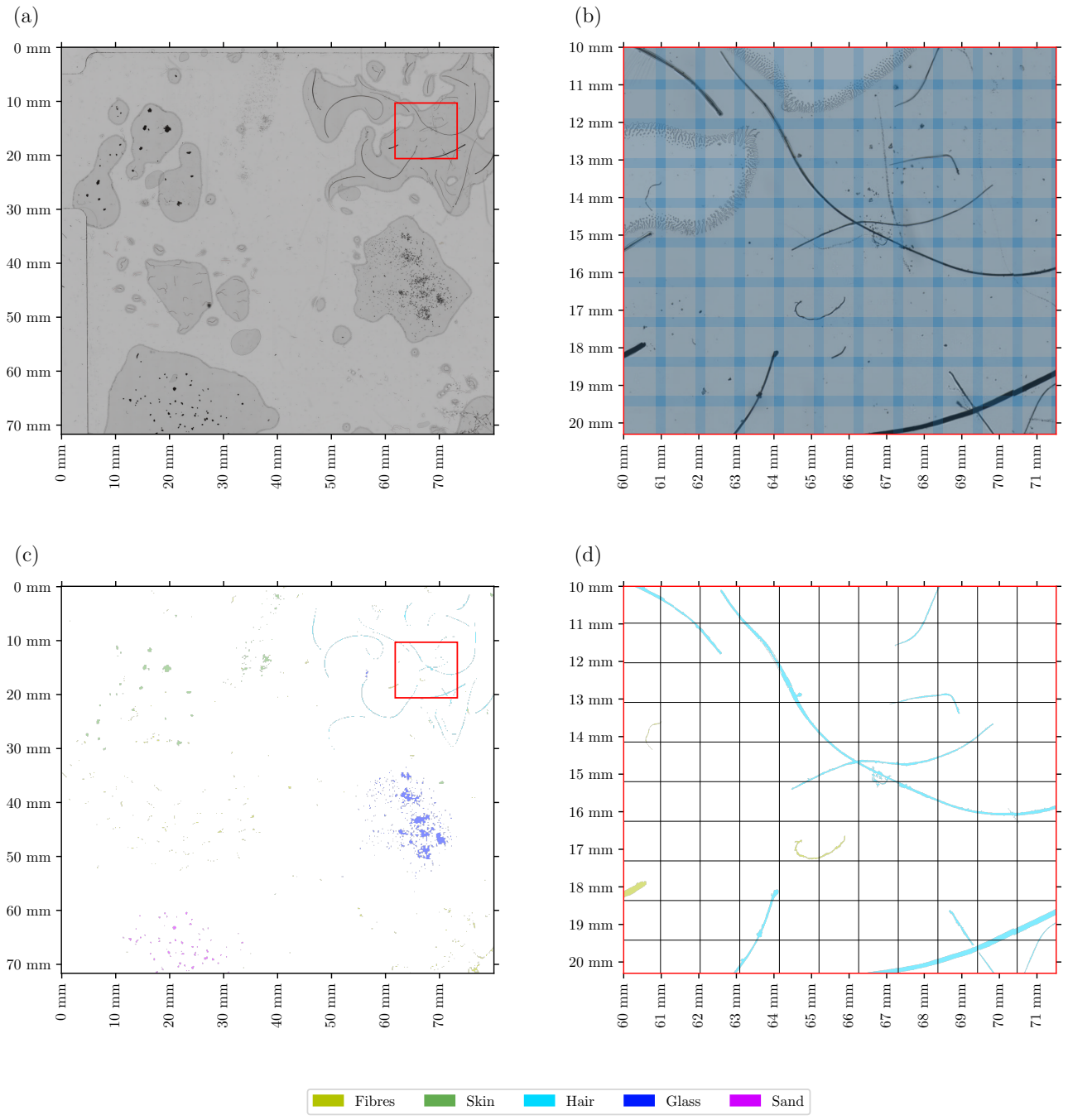


Supplementary Figure S13: Details of extracting rotated crops from microtrace scans. At a selected coordinate p, either one image is extracted for microtrace training or a pair of images is extracted for SSL pretraining. For a given base patch size $t \times t$, magnification M, rotation $\alpha \in [-\pi, \pi]$, aspect ratio AR and translation $\vec{d}$, the crop is fully determined. With the magnification, the area is determined as $\frac{t^2}{M^2} = hw$. With the aspect ratio AR$= h/w$, the patch extraction height h and width w are determined as $h = \frac{t}{M}\sqrt{\text{AR}}$, $w = \frac{t}{M}\sqrt{\frac{1}{\text{AR}}}$. For example for an aspect ratio of 4 and a magnification of 2x (+100%), this results in $h = t$ and $w = t/4$. For a rotation α, first a larger image area is extracted of height $h\,|\sin\alpha| + w\,|\cos\alpha|$ and width $w\,|\sin\alpha| + h\,|\cos\alpha|$. Then, the rotated crop is made with the affine transformation matrix that maps the points A, B, C to the points $(0,0), (t,0), (0,t)$ of the final image patch. Here, nearest neighbour interpolation is used [7]. For $\alpha < 0$, the Figure can be adjusted accordingly.

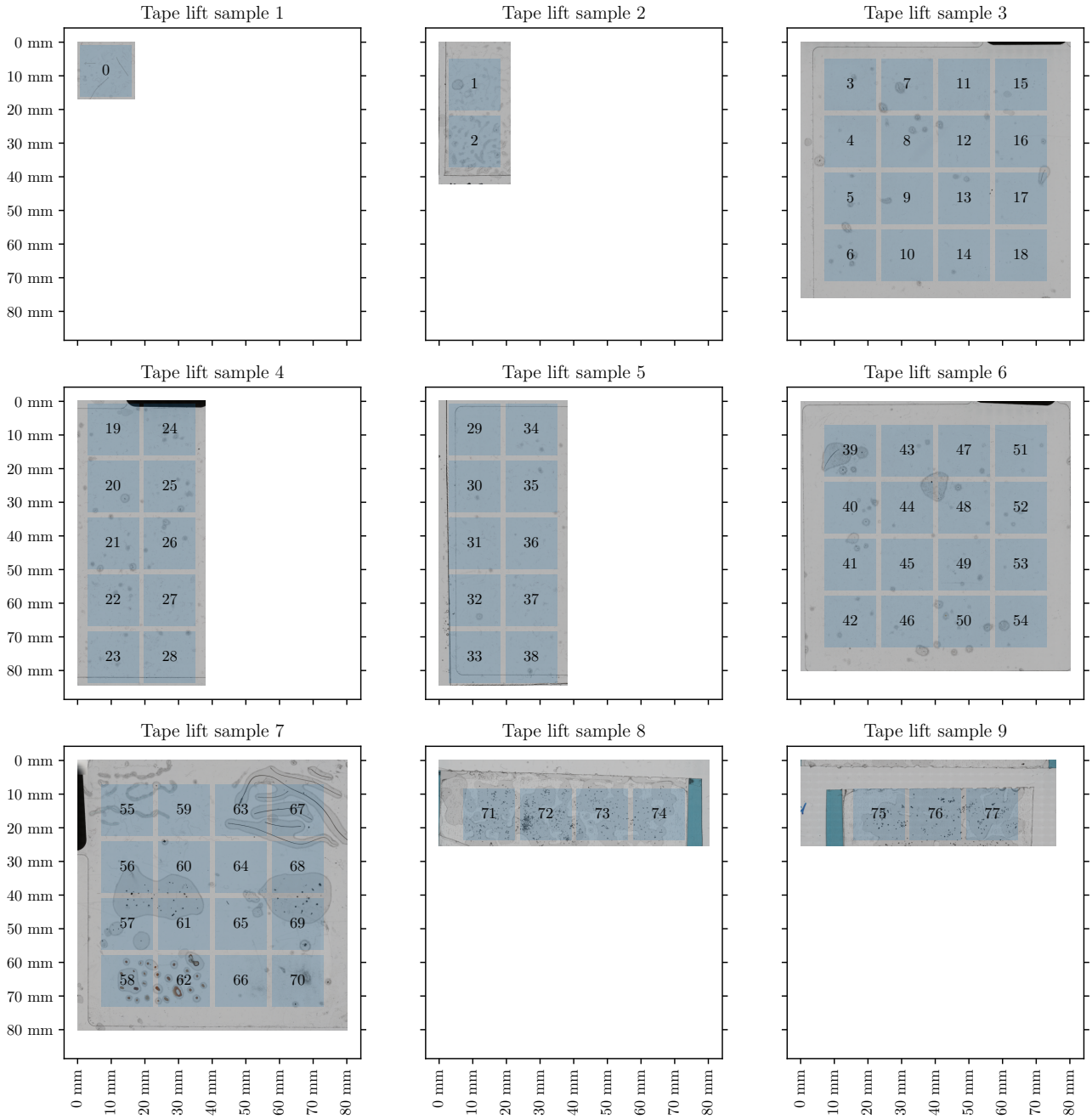




Supplementary Figure S14: Predictions for the unannotated dataset. (a) Predictions for fully random crops in the unannotated dataset. Here, the location of the crops was selected via uniform sampling. (b) Predictions for randomly selected foreground crops in the unannotated dataset. These crops were extracted with the thresholding approach discussed in Subsection IV.D.



Supplementary Figure S15: Test scan with annotations. The scan is subdivided into a uniform grid. The model takes a 1280×1280 µm image patch as input to make predictions for a patch of 1024×1024 µm and discards the predictions made for the outer pixels. (a) Overview of scan. (b) Zoomed-in view showing the image patches of 1280×1280 µm. (c) Overview of corresponding expert annotation. In this figure, the annotations have been thickened to improve visibility. (d) Zoomed-in view of annotations including the grid of 1024×1024 µm patches.



Supplementary Figure S16: Subdivision of annotated training scans into equally sized regions. Only image patches with centre points inside the blue marked regions are processed during training. The margin between the blue regions prevents the FOV of the patches from overlapping.

Supplementary Table S1: Overview of all hyperparameters. The values are discussed and motivated in Section IV.

Hyperparameter	Value
Trace training (see Subsection IV.A)	
Batch size m	160
Weight decay	$1 \cdot 10^{-4}$
SGD momentum	0.9
Base learning rate	0.016
Learning rate scheduling	$\times 0.1$ at 70th and 90th percentile
Data point steps	400k (pretrained) / 2 million (scratch)
Patch size	256×256 pixels
Image resolution	4 µm/pixel
Pixel intensity threshold of foreground T	166
Pre-processing and augmentation	see Table S2
ImageNet pretraining (see Subsection IV.B, [53])	
Batch size m	32
Weight decay	$1 \cdot 10^{-4}$
SGD momentum	0.9
Base learning rate	0.1
Learning rate scheduling	$\times 0.1$ at 33rd and 66th percentile
Data point steps	115 million
Self-supervised pretraining (see Subsection IV.C)	
Batch size m	80
Weight decay	$1.5 \cdot 10^{-6}$
SGD momentum	0.9
Base learning rate	0.125
Learning rate scheduling	linear warm-up & cosine decay
Data point steps	40 million
Base moving average parameter τ_0	0.99992
Moving average parameter scheduling	$\tau_k = 1 - (1 - \tau_0) \ (\cos(\pi k/K) + 1)/2$
Patch size	256×256 pixels
Image resolution	4 µm/pixel
Pixel intensity threshold of foreground T	166
Pre-processing and augmentation	see Table S2

Supplementary Table S2: Image transformations and augmentations. The non-cursive operations regard data augmentations, the cursive operations regard pre-processing. The operations take place in the stated order. $U(x)$ denotes a uniform random variable between $-x, x$. $F(x)$ denotes a random variable enforcing multiplicative symmetry with output $1+|u|$ for $u \geq 0$ and $1/(1+|u|)$ for $u < 0$ where $u \sim U(-x, x)$. During SSL pretraining, one of the images of each pair is augmented with set 1, the other with set 2. The grey cells in the right column have equal values to the corresponding cells in the middle column. With the exception of the viewpoint crop, all augmentations and the corresponding parameters are equal to the ones proposed in [28]. During microtrace training, no translation is used. The foreground pixels at which the image patches are centred, are sampled without replacement. This prevents two image patches being located at the same location, thus making translation augmentation redundant.

Operation	SSL set 1	SSL set 2	Microtrace training
Viewpoint crop			
Horizontal translation	$U(d_m)$	$U(d_m)$	0
Vertical translation	$U(d_m)$	$U(d_m)$	0
Magnification	$F(z_f)$	$F(z_f)$	$F(1.0)$
Aspect ratio	$F(1/3)$	$F(1/3)$	$F(1/3)$
Rotation angle	$U(\pi)$	$U(\pi)$	$U(\pi)$
Resize to $t \times t$ pixels (linear interpolation)			
Horizontal flip			
Application probability	0.5	0.5	0.5
Vertical flip			
Application probability	0.5	0.5	0.5
Colour jitter			
Application probability	0.8	0.8	0.8
Brightness adjustment	$1 + U(0.4)$	$1 + U(0.4)$	$1 + U(0.4)$
Contrast adjustment	$1 + U(0.4)$	$1 + U(0.4)$	$1 + U(0.4)$
Saturation adjustment	$1 + U(0.2)$	$1 + U(0.2)$	$1 + U(0.2)$
Hue adjustment	$1 + U(0.1)$	$1 + U(0.1)$	$1 + U(0.1)$
Colour dropping			
Application probability	0.2	0.2	0.2
Gaussian blurring			
Application probability	1.0	0.1	0.5
Kernel size	23×23	23×23	23×23
Standard deviation	$1.05 + U(0.95)$	$1.05 + U(0.95)$	$1.05 + U(0.95)$
Solarization			
Application probability	0.0	0.2	0.0
Normalisation by subtracting $\mu = 0.70$ and dividing by $\sigma = 0.05$			

Supplementary Table S3: Initialisation of networks. For the pretrained models, the transferred parameters include all trainable parameters, including batch normalisation statistics. The descriptions of the network blocks are given in Supplementary Figure S8 and S9.

Block	Initialisation
Microtrace training	
ResNet-50	SSL Pretrained / Random [31] / IMAGENET1K_V1 [17]
Stage $6-7$	Random [31]
1×1 convolutions	Random [31]
Transpose convolutions	Bilinear upsamplers [67]
SSL Pretraining	
Online ResNet-50	Random [31] / IMAGENET1K_V1 [17]
Online Projector MLP	Random [31]
Online Predictor MLP	Random [31]
Target ResNet-50	Copy of online ResNet-50 weights
Target Projector MLP	Copy of online projector weights

Supplementary Table S4: Comparison of class distribution in annotated and unannotated dataset. For the unannotated dataset, the predictions of the model presented in Subsection II.A are taken.

Class	Relative label occurrence (‰)	
	Annotated dataset (expert labels)	Unannotated dataset (model predictions)
Fibres	4.4	3.2
Skin	6.0	2.1
Hair	0.38	1.3
Glass	0.18	1.3
Sand	0.08	0.54
Background	988.9	991.7

Chapter 4

Conclusion

Forensic microtraces, such as textile fibres, glass particles, hairs, and skin cells, provide crucial information in criminal investigations. Finding these traces is a time-consuming process that heavily relies on manual labour. Amongst other methods, experts recover traces with one-to-one taping. Here, a large number of transparent adhesive tapes is applied to the surface area of a body or an object. These tapes are subsequently analysed by experts through manual microscopy. As the traces are scattered across the tapes and the total taped area can span more than a square meter [18], this yields a labour-intensive process, causing it to be applied only in few cases. In most cases, trace investigation is therefore limited to selected traces and regions of interest [18, 51] or not applied at all [56]. On the other hand, the development of automated microscopy, such as [70], allows tape lift samples to be scanned automatically. This paves the way for the application of computer vision models to automate the trace-finding process and decrease trace investigation costs, allowing it to be applied more often and more rigorously.

In this work a deep learning model is proposed for semantic segmentation of forensic microtraces in microscopy images. This model allows trace classification and localisation through pixel-wise labelling.

As the development of such a model requires extensive manual labour in labelling training data, the benefit of pretraining is investigated. Three pretraining configurations are tested and compared to training from scratch. Firstly, ImageNet pretrained models are tested. These widely available models that are pretrained through classification of everyday photographs. Secondly, we test pretraining with self-supervised learning (SSL) on unannotated microtrace images. Here, the network is trained to extract similar feature representations for two artificial modifications (augmentations) of an unknown microtrace image. Thirdly, we test a combined approach in which we apply SSL-pretraining on a ImageNet-pretrained model.

The trace recognition ability of the pretrained models is evaluated with the mean Intersection over Union (mIoU). Combining ImageNet pretraining with SSL pretraining is found to result in the highest mIoU. With this pretraining approach, hairs, fibres, skin and glass are recognised in microscopy images with an mIoU of 0.56 while only being trained on 2.2 dm^2 of annotated tape lift scans. On the other hand, a model trained from scratch on the same amount of annotated data achieves an mIoU of 0.34. This can be outperformed with pretraining even if the available amount of annotated data is reduced four-fold to 0.6 dm^2, with which an mIoU of 0.36 is achieved.

As tape lift scans are largely composed of background, an image extraction method that leverages thresholding is proposed. This allows the foreground to be oversampled, which results in a 21% increase in mIoU compared to extracting images uniformly from the scans.

The results of this thesis provide a foundation for future research. Opportunities include exploring other augmentation strategies, analysis and expansion of the datasets and possible alternative microscopy imaging modes, alternative neural network architectures, incorporating instance segmentation to count traces and extending the classification to a broader range of trace classes and subclasses. Lastly, further research can be done in advancing the benefit of SSL pretraining through dense, segment-level comparison of image pairs.

Altogether, this work makes four significant contributions. Firstly, a new baseline for microtrace recognition on tape lift scans is set. Secondly, the benefit of SSL pretraining for label-efficient learning is investigated. Thirdly, an image extraction method is proposed to efficiently learn on background-rich microscopy scans. Collectively, these contributions make a significant step in automating the microtrace finding process. This advances trace investigation and ultimately contributes to improving our justice system.

Appendix A

Shading correction

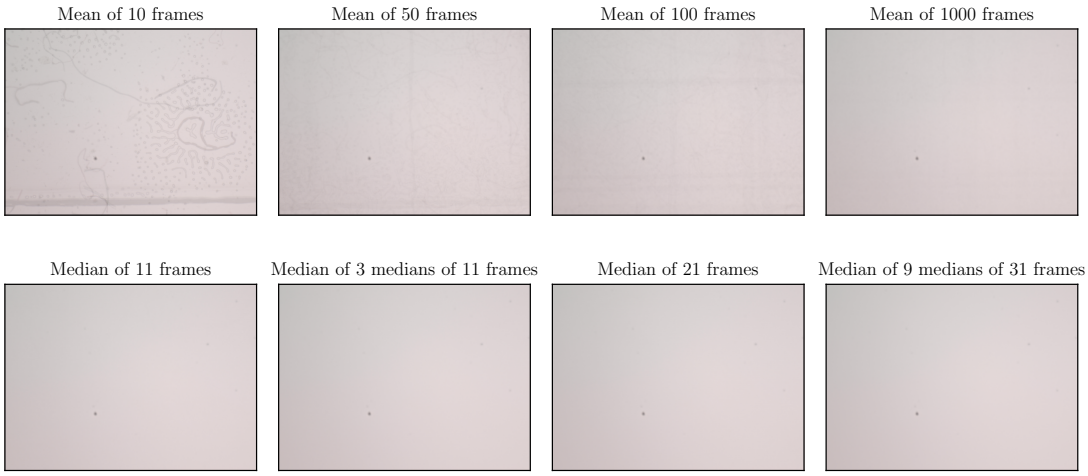
In Figure 2-4a, it was shown that captured frames of automated microscopy scans can contain a non-uniform illumination that visually degrades the scan. To correct this, a shading correction filter was designed.

Shading correction methods can be categorised as prospective or retrospective [55]. Prospective methods use separate calibration images, while retrospective methods estimate suitable filtering after the images are acquired and no calibration image is available. The scans used in this thesis project were generated in various imaging settings over the time-span of a few months. As calibration images for each of the settings were unavailable, a retrospective filtering method was designed.

The filter starts with estimating the background image X_{bg} that the camera would have captured for a frame without objects. Here, the large background ratio of the images is used (see Table 1 and Figure S12). As objects occupy only a small area of the camera frames, most frames will primarily consist of background. Assuming that the objects are located randomly within each frame, each camera pixel at position i_x, i_y on the sensor has a much larger probability of seeing background than seeing an object. Thus, when considering a large number of frames, each pixel at location i_x, i_y will see the background X_{bg} for the majority of the frames. Assuming that each pixel sees background for at least 50 % of the frames, X_{bg} can be estimated by taking the median of each pixel at location i_x, i_y over a set of sampled frames.

Taking the pixel-wise median is compared to taking the pixel-wise mean in Figure A1. Here, it can be seen that the median approach requires fewer images to properly estimate X_{bg}. The image obtained by taking the mean over 10 camera frames still contains clear artefacts of traces. However, it should be noted that calculating the median for a large set of frames draws large memory requirements and can be computationally demanding [47]. For a set of n frames, the average can be computed by consecutively multiplying

each frame by $1/n$ and writing the result of the summation of frames up to that frame into memory. Thus, only two memory locations are required. Calculating the median on the other hand requires storing and sorting the pixel values of all n frames, resulting in a computationally expensive operation. Therefore, Figure A1 regards the median of the median to approximate the full-range median [9, 47] for large n.



Supplementary Figure A1: Comparison of estimating background image X_{bg} for shading correction via pixel-wise mean and median. The used frames are sampled randomly from a single scan. It can be seen that for both the mean and median the outline of foreground objects fade out from the image. Furthermore, the figure shows that the median-based approach converges faster than the mean-based approach. The converged frames in the right column show a brighter lightning in the right halves than in the left halves. Furthermore, a dirt spot can be seen in the bottom-left quarters of the frames.

X_{bg} is chosen to be estimated by taking the median of 3 medians of 21 frames, requiring a total of 63 camera frames to estimate the calibration image for each scan. The median is calculated separately for each colour channel.

Now, X_{bg} is used to determine a correction factor. In general, an image captured with transmission microscopy can be approximated linearly as [55, 78]:

$$I_{\text{meas}}\left(i_x, i_y, i_c\right) = I_{\text{true}}\left(i_x, i_y, i_c\right) \times S(i_x, i_y, i_c) + D(i_x, i_y, i_c). \tag{1}$$

Here, $I_{\text{meas}}\left(i_x, i_y, i_c\right)$ represents the measured image, while $I_{\text{true}}\left(i_x, i_y, i_c\right)$ represents the uncorrupted, shading-free counterpart. $S(i_x, i_y, i_c), D(i_x, i_y, i_c)$ equal the *flat-field* and *dark-field* term respectively. Flat-field relates to the variety in illumination across the image, while dark-field relates to camera offset and thermal noise [55].

A shading-free image without objects consists of a single, homogeneous, background colour $C_{\text{bg}}(i_c)$. This forms $I_{\text{true}}(i_x, i_y, i_c) = C_{\text{bg}}(i_c)$. As described earlier, $I_{\text{meas}} = X_{\text{bg}}$

is used to estimate the measured image for recording a frame without objects. We now neglect the dark-field term D, which allows $S(i_x, i_y, i_c)$ to be calculated from:

$$S(i_x, i_y, i_c) = \frac{X_{\text{bg}}(i_x, i_y, i_c)}{I_{\text{true}}(i_x, i_y, i_c)} = \frac{X_{\text{bg}}(i_x, i_y, i_c)}{C_{\text{bg}}(i_c)}. \tag{2}$$

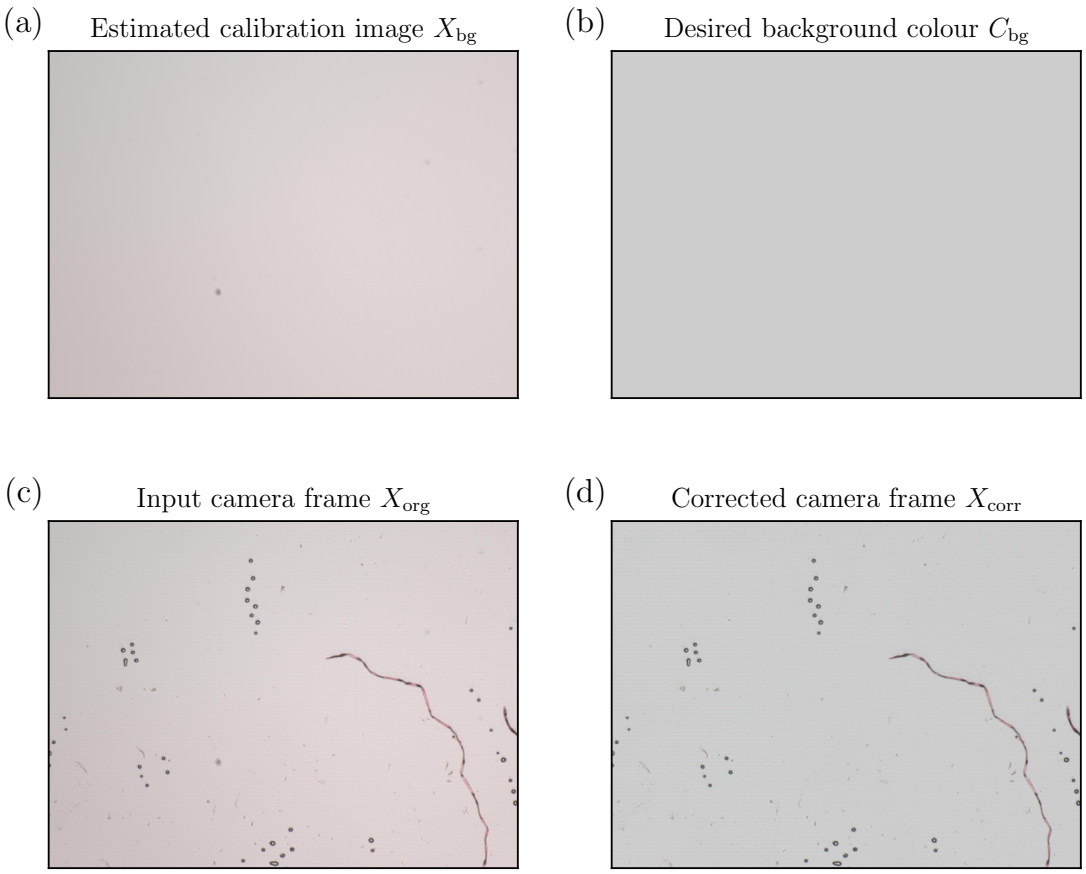
Where the colour C_{bg} is chosen as RGB(179,179,179). This is based on the average pixel intensity that was encountered in 10 randomly selected scans, although it can be adjusted based on brightness preferences. After calculating $S(i_x, i_y, i_c)$ for each channel of each camera frame pixel, images can be corrected as:

$$X_{\text{corr}}(i_x, i_y, i_c) = \frac{X_{\text{org}}(i_x, i_y, i_c)}{S(i_x, i_y, i_c)}. \tag{3}$$

Here, X_{org} is the uncorrected image and X_{corr} the corrected counterpart. An example of a corrected scan is given in Figure 2-4b. It can be seen that the designed filter is able to homogenise the background and remove the shading pattern of the individual camera frames.

A dark spot can be distinguished in the estimated X_{bg} shown in Figure A2a. This was caused by dirt on one of the lenses. According to the law of Beer-Lambert (2-1), the absorbtion of light by the dirt particle results in a multiplicative effect on measured illumination. This yields (1) and thereby (3) valid, as $D(i_x, i_y, i_c)$ was neglected. As a result, the filter is able to remove the appearance of the dirt particle from the camera frames. This is shown in Figure A2d.

For applications in which the dark-field term can not be neglected, alternative approaches might be suitable [55, 78]. However, the increased complexity of these approaches can bear additional computational requirements.



Supplementary Figure A2: Correction of a randomly selected single camera frame. It can be seen that the uncorrected image (c) has a darker illumination on the left side than on the right side, while the corrected image (d) has an even illumination. Moreover, the dirt spot on the camera that can clearly be seen in (a) is filtered out in the corrected version. See also Figure 2-4 for a visualisation of the whole tape lift scan.

Bibliography

[1] Saleh Albelwi. Survey on Self-Supervised Learning: Auxiliary Pretext Tasks and Contrastive Learning Methods in Imaging. *Entropy*, 24(4), 2022.

[2] Arian Bakhtiarnia, Qi Zhang, and Alexandros Iosifidis. Efficient High-Resolution Deep Learning: A Survey, 7 2022.

[3] Konstantinos Balas. Trace Microanalysis Microscope Systems and Methods. Technical report, World Intellectual Property Organization, 5 2021.

[4] Randall Balestriero, Mark Ibrahim, Vlad Sobal, Ari Morcos, Shashank Shekhar, Tom Goldstein, Florian Bordes, Adrien Bardes, Gregoire Mialon, Yuandong Tian, and others. A Cookbook of Self-Supervised Learning. *arXiv preprint arXiv:2304.12210*, 2023.

[5] Peter Bankhead, Maurice B. Loughrey, José A. Fernández, Yvonne Dombrowski, Darragh G. McArt, Philip D. Dunne, Stephen McQuaid, Ronan T. Gray, Liam J. Murray, Helen G. Coleman, Jacqueline A. James, Manuel Salto-Tellez, and Peter W. Hamilton. QuPath: Open source software for digital pathology image analysis. *Scientific Reports*, 7(1), 12 2017.

[6] C Bishop, M Jordan, J Kleinberg, and B Schölkopf. *Pattern Recognition and Machine Learning*, volume 4. Springer, 2006.

[7] G Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.

[8] Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. Coco-stuff: Thing and stuff classes in context. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1209–1218, 2018.

[9] T Calvo and R Mesiar. Generalized medians. *Fuzzy Sets and Systems*, 124(1):59–64, 2001.

[10] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2017.

[11] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 801–818, 2018.

[12] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607, 2020.

[13] Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey Hinton. Big Self-Supervised Models are Strong Semi-Supervised Learners. *Advances in neural information processing systems*, 33:22243–22255, 6 2020.

[14] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15750–15758, 2021.

[15] Ozan Ciga, Tony Xu, and Anne Louise Martel. Self supervised contrastive learning for digital histopathology. *Machine Learning with Applications*, 7:100198, 2022.

[16] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016.

[17] T Coyle, A Larkin, K Smith, S Mayo, A Chan, and N Hunt. Fibre mapping – a case study. *Science & Justice*, 44(3):179–186, 2004.

[18] K De Wael, L Lepot, K Lunstroot, and F Gason. 10years of 1:1 taping in Belgium — A selection of murder cases involving fibre examination. *Science & Justice*, 56(1):18–28, 2016.

[19] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[20] Nilanjan Dey. Uneven illumination correction of digital images: A survey of the state-of-the-art. *Optik*, 183:483–495, 2019.

[21] J E Dobson. *The Birth of Computer Vision*. University of Minnesota Press, 2023.

[22] Carl Doersch, Abhinav Gupta, and Alexei A Efros. Unsupervised Visual Representation Learning by Context Prediction. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 1422–1430, 2015.

[23] M Everingham, L Van˜Gool, C K I Williams, J Winn, and A Zisserman. The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results. http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html.

[24] M Everingham, L Van˜Gool, C K I Williams, J Winn, and A Zisserman. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html.

[25] Yanbo Feng, Adel Hafiane, and Hélène Laurent. A deep learning based multiscale approach to segment the areas of interest in whole slide images. *Computerized Medical Imaging and Graphics*, 90:101923, 2021.

[26] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised Representation Learning by Predicting Image Rotations. *CoRR*, abs/1803.07728, 2018.

[27] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016.

[28] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, and others. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 33:21271–21284, 2020.

[29] Matthew G Hanna, Anil Parwani, and Sahussapont Joseph Sirintrapun. Whole slide imaging: technology and applications. *Advances in Anatomic Pathology*, 27(4):251–259, 2020.

[30] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020.

[31] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.

[32] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[33] Gabriel Huang, Issam Laradji, David Vazquez, Simon Lacoste-Julien, and Pau Rodriguez. A survey of self-supervised and few-shot object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[34] Haris Iqbal. HarisIqbal88/PlotNeuralNet v1.0.0, 12 2018.

[35] Yoshimasa Kawazoe, Kiminori Shimamoto, Ryohei Yamaguchi, Yukako Shintani-Domoto, Hiroshi Uozaki, Masashi Fukayama, and Kazuhiko Ohe. Faster R-CNN-based glomerular detection in multistained human whole slide images. *Journal of Imaging*, 4(7), 12 2018.

[36] Yejin Kim and Tae Sup Yun. How to classify sand types: A deep learning approach. *Engineering geology*, 288:106142, 2021.

[37] Simon Kornblith, Jonathon Shlens, and Quoc V Le. Do better imagenet models transfer better? In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2661–2671, 2019.

[38] K Koutroumbas and S Theodoridis. *Pattern Recognition*. Elsevier Science, 2008.

[39] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. ImageNet Classification with Deep Convolutional Neural Networks. In F Pereira, C J Burges, L Bottou, and K Q Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012.

[40] Emma A Levin, Ruth M Morgan, Lewis D Griffin, and Vivienne J Jones. A comparison of thresholding methods for forensic reconstruction studies using fluorescent powder proxies for trace materials. *Journal of forensic sciences*, 64(2):431–442, 2019.

[41] Hao Li, Zheng Xu, Gavin Taylor, Christoph Studer, and Tom Goldstein. Visualizing the loss landscape of neural nets. *Advances in neural information processing systems*, 31, 2018.

[42] Ziqiang Li, Rentuo Tao, Qianrun Wu, and Bin Li. DA-RefineNet:A Dual Input Whole Slide Image Segmentation Algorithm
}Based on Attention. *CoRR*, abs/1907.06358, 2019.

[43] X Liu, F Zhang, Z Hou, L Mian, Z Wang, J Zhang, and J Tang. Self-Supervised Learning: Generative or Contrastive. *IEEE Transactions on Knowledge & Data Engineering*, 35(01):857–876, 1 2023.

[44] Ilya Loshchilov and Frank Hutter. SGDR: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.

[45] Niall O' Mahony, Sean Campbell, Anderson Carvalho, Suman Harapanahalli, Gustavo Adolfo Velasco-Hernández, Lenka Krpalkova, Daniel Riordan, and Joseph Walsh. Deep Learning vs. Traditional Computer Vision. *CoRR*, abs/1910.13796, 2019.

[46] Michael Majurski and Peter Bajcsy. Exact tile-based segmentation inference for images larger than gpu memory. *Journal of Research of the National Institute of Standards and Technology*, 126:1–16, 2021.

[47] Patrenahalli M Narendra. A Separable Median Filter for Image Noise Smoothing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-3(1):20–29, 1981.

[48] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VI*, pages 69–84, 2016.

[49] Marin Oršić and Siniša Šegvić. Efficient semantic segmentation with pyramidal fusion. *Pattern Recognition*, 110:107611, 2021.

[50] Rafael Padilla, Wesley L Passos, Thadeu L B Dias, Sergio L Netto, and Eduardo A B da Silva. A Comparative Analysis of Object Detection Metrics with a Companion Open-Source Toolkit. *Electronics*, 10(3), 2021.

[51] Ray Palmer. A 'Zonal' Approach to Fibre Recovery at Scenes. In *Proceedings of the 14th European Fibre Group Meeting*, 2006.

[52] Dim P Papadopoulos, Ethan Weber, and Antonio Torralba. Scaling up instance annotation via label propagation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15364–15373, 2021.

[53] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, and others. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

[54] Chaitanya Patel, Shashank Sharma, Valerie J Pasquarella, and Varun Gulshan. Evaluating self and semi-supervised methods for remote sensing segmentation tasks. *arXiv preprint arXiv:2111.10079*, 2021.

[55] Tingying Peng, Kurt Thorn, Timm Schroeder, Lichao Wang, Fabian J Theis, Carsten Marr, and Nassir Navab. A BaSiC tool for background and shading correction of optical microscopy images. *Nature communications*, 8(1):14836, 2017.

[56] Joseph L Peterson, Matthew J Hickman, Kevin J Strom, and Donald J Johnson. Effect of forensic evidence on criminal justice case processing. *Journal of forensic sciences*, 58:S78–S90, 2013.

[57] Narinder Singh Punn and Sonali Agarwal. BT-Unet: A self-supervised learning framework for biomedical image segmentation using barlow twins with U-net models. *Machine Learning*, pages 1–16, 2022.

[58] Colorado J Reed, Xiangyu Yue, Ani Nrusimha, Sayna Ebrahimi, Vivek Vijaykumar, Richard Mao, Bo Li, Shanghang Zhang, Devin Guillory, Sean Metzger, and others. Self-supervised pretraining improves self-supervised pretraining. In *Proceedings*

of the IEEE/CVF Winter Conference on Applications of Computer Vision, pages 2584–2594, 2022.

[59] Bernard Robertson, G A Vignaux, and Charles Berger. *Interpreting Evidence: Evaluating Forensic Science in the Courtroom: Second Edition.* Wiley, 3 2016.

[60] Robertson James, Claude Roux, and Kenneth G Wiggins. *Forensic Examination of Fibres.* CRC Press, 2021.

[61] Ingrid C Romero, Shu Kong, Charless C Fowlkes, Carlos Jaramillo, Michael A Urban, Francisca Oboh-Ikuenobe, Carlos D'Apolito, and Surangi W Punyasena. Improving the taxonomy of fossil pollen using convolutional neural networks and superresolution microscopy. *Proceedings of the National Academy of Sciences*, 117(45):28496–28505, 2020.

[62] Ronneberger Olafand Fischer and Philippand Brox Thomas. U-Net: Convolutional Networks for Biomedical Image Segmentation. In Joachim Navab Nassir }and Hornegger, Wells William M., and Frangi Alejandro F, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pages 234–241, Cham, 2015. Springer International Publishing.

[63] Paul L Rosin. Unimodal thresholding. *Pattern Recognition*, 34(11):2083–2096, 2001.

[64] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.

[65] Tom G Schotman and Jaap van der Weerd. On the recovery of fibres by tape lifts, tape scanning, and manual isolation. *Science & Justice*, 55(6):415–421, 2015.

[66] Tom G Schotman, Xiaoma Xu, Nicole Rodewijk, and Jaap van der Weerd. Application of dye analysis in forensic fibre and textile examination: Case examples. *Forensic Science International*, 278:338–350, 2017.

[67] Evan Shelhamer, Jonathan Long, Trevor Darrell, and others. Fully convolutional networks for semantic segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 39(4):640–651, 2017.

[68] Kelly J Sheridan, Ray Palmer, David A Chalton, Jariel N Bacar, Jack Beckett, Kieran Bellerby, Lucy Brown, Emily Donaghy, Alexander Finlayson, Cameron Graham, Beth Robertson, Lauren Taylor, and Matteo D Gallidabino. A quantitative assessment of the extent and distribution of textile fibre transfer to persons involved in physical assault. *Science & Justice*, 63(4):509–516, 2023.

[69] Saeed Shurrab and Rehab Duwairi. Self-supervised learning methods and applications in medical imaging analysis: A survey. *PeerJ Computer Science*, 8:e1045, 2022.

[70] Shuttle PCP. Scientific High-throughput and Unified Toolkit for Trace analysis by forensic Laboratories in Europe.

[71] Greenfield Sluder and David Wolf. *Digital Microscopy*. Elsevier, 4 edition, 2013.

[72] T. Biermann. Collection of fibres on a 1:1 scale. In *Proceedings of the 6th European Fibre Group Meeting, Dundee*, pages 44–47, 1998.

[73] Yonglong Tian, Chen Sun, Ben Poole, Dilip Krishnan, Cordelia Schmid, and Phillip Isola. What makes for good views for contrastive learning? *Advances in neural information processing systems*, 33:6827–6839, 2020.

[74] Yuandong Tian, Xinlei Chen, and Surya Ganguli. Understanding self-supervised learning dynamics without contrastive pairs. In *International Conference on Machine Learning*, pages 10268–10278, 2021.

[75] Yuandong Tian, Xinlei Chen, and Surya Ganguli. Understanding self-supervised learning dynamics without contrastive pairs. In *International Conference on Machine Learning*, pages 10268–10278, 2021.

[76] Xinlong Wang, Rufeng Zhang, Chunhua Shen, Tao Kong, and Lei Li. Dense contrastive learning for self-supervised visual pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3024–3033, 2021.

[77] Yi Wang, Conrad M Albrecht, Nassim Ait Ali Braham, Lichao Mou, and Xiao Xiang Zhu. Self-Supervised Learning in Remote Sensing: A review. *IEEE Geoscience and Remote Sensing Magazine*, 10(4):213–247, 2022.

[78] Qiang Wu, Fatima A Merchant, and Kenneth R Castleman. *Microscope Image Processing*. Academic Press, 4 2008.

[79] Jiang Xiaojia, Yang Mengjing, Quan Yongzhi, and He Ya. Hair microscopic image classification method based on convolutional neural network. In *2019 IEEE International Conference on Power, Intelligent Computing and Systems (ICPICS)*, pages 433–438, 2019.

[80] Zhenda Xie, Yutong Lin, Zheng Zhang, Yue Cao, Stephen Lin, and Han Hu. Propagate yourself: Exploring pixel-level consistency for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16684–16693, 2021.

[81] Hongwei Yong, Jianqiang Huang, Deyu Meng, Xiansheng Hua, and Lei Zhang. Momentum batch normalization for deep learning with small batch size. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XII 16*, pages 224–240, 2020.

[82] Gregory W Zack, William E Rogers, and Samuel A Latt. Automatic measurement of sister chromatid exchange frequency. *Journal of Histochemistry & Cytochemistry*, 25(7):741–753, 1977.

[83] Syed Sahil Abbas Zaidi, Mohammad Samar Ansari, Asra Aslam, Nadia Kanwal, Mamoona Asghar, and Brian Lee. A survey of modern deep learning based object detection models. *Digital Signal Processing*, page 103514, 2022.

[84] H Zhao, J Shi, X Qi, X Wang, and J Jia. Pyramid Scene Parsing Network. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6230–6239, Los Alamitos, CA, USA, 7 2017. IEEE Computer Society.

[85] Yuanyuan Zhou, Ji Zhang, Jiao Huang, Kaifei Deng, Jianhua Zhang, Zhiqiang Qin, Zhenyuan Wang, Xiaofeng Zhang, Ya Tuo, Liqin Chen, and others. Digital whole-slide image analysis for automated diatom test in forensic cases of drowning using a convolutional neural network algorithm. *Forensic science international*, 302:109922, 2019.

[86] Janina Zięba-Palus. Forensic examinations of micro-traces. *Forensic Science and Criminology*, 2(2), 2017.

[87] Moritz Zink, Martin Schiele, Pengcheng Fan, and Stephan Gasterstädt. Boosting Mask R-CNN Performance for Long, Thin Forensic Traces with Pre-Segmentation and IoU Region Merging. *arXiv preprint arXiv:2203.03886*, 2022.

Glossary

List of Acronyms

WSI	whole slide imaging
NFI	Netherlands Forensic Institute
SGD	Stochastic Gradient Descent
IoU	Intersection over Union
MLP	multilayer perceptron
mIoU	mean Intersection over Union
SSL	self-supervised learning
FOV	field of view

List of Symbols

General mathematics

$\cap$	Intersection
$\cup$	Union
∇_θ	Gradient with respect to θ
$p(Y \mid X)$	Probability of Y under the condition X
$\#$	Count of

Digital image processing

α	Rotation angle
h	Image region height

M	Magnification
T	Threshold
t	Tile size
w	Image region width
AR	Aspect ratio

Machine learning

ϵ	Constant for numerical stability
η	Learning rate
$\hat{Y}$	Model prediction
$\mathcal{L}$	Cost function
σ	Activation function
τ	Exponential moving average parameter
θ, ξ	Model weights
f	Model architecture
f_θ	Trained model
K	Kernel
k	Optimisation step
m	Batch size

Microscopy

λ	Wavelength
ε	Absorption coefficient of material
d	Length of light path through sample
$I(\lambda)$	Light intensity at wavelength λ
$I(p)$	Light intensity of pixel p