Deep learning on higher harmonic generation images for regression and pathology

Siem de Jong

30-6-2023

Deep learning on higher harmonic generation images for regression and pathology

Siem de Jong

Faculty of Science, University of Amsterdam
Faculty of Science, Vrije Universiteit Amsterdam
LaserLaB Amsterdam, *Biophotonics and Biomedical Imaging*

Report Master Project Physics and Astronomy
track Biophysics and Biophotonics
60 EC
Conducted between 05-09-2022 and 30-06-2023

Daily supervisor	Dr. rer. nat. P. J. González
Examiner	Prof. dr. M. L. Groot
Second reviewer	Dr. rer. nat. D. W. A. Hillmann

30-06-2023

Colophon
This document was typeset with the help of KOMA-Script and LaTeX using the kaobook class. The source code is published on siemdejong/mscthesis.

Cover: An AEQVIPEDIS transformed higher harmonic generation microscopy image of a pediatric brain tumor.

Abstract

Higher harmonic generation (HHG) microscopy allows for imaging of biological tissue at subcellular resolution. At this scale, mechanical skin properties or brain tumor presence may be observed. Clinical settings need artificial intelligence replace time-consuming measurements and human observations. For example, obtaining stress-strain curves from skin tissue requires mechanical measurements, and intraoperatively diagnosing tumors requires clinicians to inspect images thoroughly. To predict stress-strain curves from HHG skin tissue images and distinguish two pediatric brain tumors, two convolutional neural networks are developed and validated.

The skin stress-strain curve predictor achieved a mean R^2 of -0.36 (SE 0.60). The brain tumor model could distinguish medulloblastoma from pilocytic astrocytoma with a mean average precision of 0.89 (SE 0.05) and 0.41 (SE 0.20) AUPRG.

Both models need further training and external validation. After additional training and validation, updated models may ultimately be used to analyze live microendoscope images to be used by plastic surgeons, or to intraoperatively discriminate between pediatric patients with pilocytic astrocytoma or medulloblastoma, or to pre-select interesting regions for diagnosis.

Contents

List of Figures

List of Tables

1
General introduction

1.1 Deep learning for higher harmonic microscopy

Visualizing living tissue and cells is of vital importance in life sciences and health care. Standard, non-invasive techniques such as magnetic resonance imaging, ultrasound imaging, and computed tomography fail to image structures at resolutions high enough to distinguish structures as individual cells or connective tissue. These structures are interesting for pathologists or skin stretch experts. Higher harmonic generation (HHG) microscopy can image cells and tissue at resolutions of 0.2 µm per pixel (mpp) in seconds. These high resolution images can contain complex structures and features.

Collagen and elastin fibers are such complex structures. They are important for determining stretch properties of skin tissue. Skin tissue can be mechanically stretched to get stress-strain curves, but it is time-expensive, could break the tissue, and requires *ex vivo* measurements. Tissue images may have all information needed to determine stretch properties such as Young's modulus or maximum stress. Chapter 3 studies the possibility of acquiring stress-strain curves from second harmonic generation (SHG) images alone. This may be a step forward to find out skin properties *in vivo* with an endoscope to aid plastic surgery.

For pathology, disease patterns consist of a combination of features. Current clinical practice includes analysis of histopathological data. However, making this data takes a long time, mainly caused by tissue processing. HHG imaging can do this in seconds, allowing for intraoperative feedback. Feedback can *e.g.* include amount of resected tumor tissue or tumor type. This would still require intraoperative image analysis, while time is scarce. Chapter 4 studies the possibility to classify two pediatric brain tumors, medulloblastoma and pilocytic astrocytoma, from HHG images and explaining which regions were important for the classifications.

The experiments are preceded by an introduction on HHG imaging and deep learning concepts in Chapter 2. Chapter 5 discusses overarching challenges and gives recommendations for advancing AI for HHG imaging.

1.2 Reporting of clinical artificial intelligence

The prediction models described in this work may eventually aid health care providers in acquiring clinically relevant parameters or estimating an outcome. The Transparent Reporting of a multivariable prediction model for Individual Prognosis Or Diagnosis (TRIPOD) Initiative developed guidelines to report on such diagnostic models [1–3]. Recent advances in artificial intelligence (AI) apply AI as black box predictive models in health care, often not sufficiently well reported. Transparent reporting on these black box models builds confidence in using and further developing the models. This is especially important in health care, where there is a need for automation while trust in AI is yet to be earned. The TRIPOD statement in its current form is not well-suited for AI prediction models. The main challenges are with how models are trained and how models can explain themselves, which is often overlooked. Unlike machine learning

[1]: Collins et al. (2015), *Transparent Reporting of a multivariable prediction model for Individual Prognosis Or Diagnosis (TRIPOD): The TRIPOD Statement*
[2]: Moons et al. (2015), *Transparent reporting of a multivariable prediction model for individual prognosis or diagnosis (TRIPOD): Explanation and elaboration*
[3]: Heus et al. (2020), *Transparent reporting of multivariable prediction models in Journal and conference abstracts: Tripod for abstracts*

(ML) models, AI models learn by recognizing patterns. These patterns are then used in inference to make a prediction, possibly of clinical value. A clinician should then be explained how the model came to its conclusion, along with its confidence. To account for these challenges, an extension for the TRIPOD statement, TRIPOD-AI is currently being developed [4, 5]. Reports on the diagnostic models developed in this study aim to adhere to TRIPOD-AI as well as possible[1].

[4]: Collins et al. (2021), *Protocol for development of a reporting guideline (TRIPOD-AI) and risk of bias tool (PROBAST-AI) for Diagnostic and prognostic prediction model studies based on Artificial Intelligence*
[5]: Collins et al. (2020), *TRIPOD-AI*

[1] The reader is invited to use the TRIPOD-AI accompanying PROBAST-AI [4, 6, 7] checklist to assess the risk of bias of the predictive models.

2

Theory of higher harmonic generation and artificial neural networks

2.1 Higher harmonic generation microscopy

Higher harmonic generation is a nonlinear scattering process resulting from light interacting with tissue. Photons from the incident laser beam combine into one photon via a virtual state, preserving the energy. In this study, two higher harmonic generation variants are used: second and third harmonic generation (SHG and THG, respectively). THG happens at structural interfaces, making it useful to image *e.g.* cells and their nuclei, or axons. SHG is generated by non-centrosymmetric structures, such as collagen and microtubules. Some molecules fluoresce, producing a photon with slightly less energy than the incoming photons. The difference in energy goes into non-radiative processes, including vibrational relaxation. Fluorescence requiring two incoming photons (two-photon excitation fluorescence, 2PEF) is produced by elastin and cellular fluorophores, which autofluoresce. Figure 2.1 shows a Jablonski diagram for THG, SHG and 2PEF.

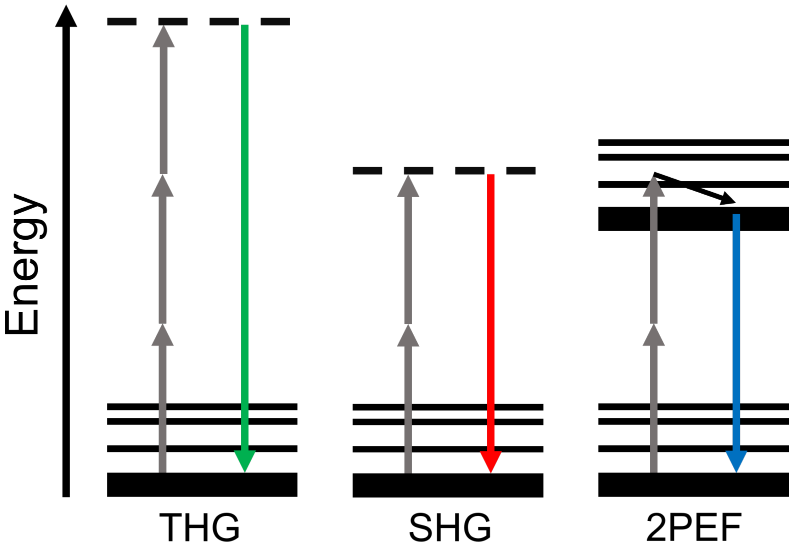


Figure 2.1: Jablonski diagrams for third and second harmonic generation (THG and SHG, respectively), and two-photon excitation fluorescence (2PEF).

2.2 From biophysics to computer science

In 1981, Hubel and Wiesel were awarded the Nobel Prize of Medicine for their discovery of visual perception [8]. They built and tested a model that describes the path of a message from eye to brain. Simply put, the message is passed on from neuron to neuron (Figure 2.2), with each neuron compiling the full message from message components. Lastly, the message is stored into the brain.

Inspired by this biological process, artificial neural networks have been developed. Later, Fukushima [10] mimicked neural networks for two-dimensional information, using convolution operations. The approach of Fukushima was inefficient. It could not learn to identify reoccurring features. To enable learning, Rumelhart, Hinton, and Williams [11] developed backpropagation: an algorithm to optimize a model to learn general features. Le Cun et al. [12] were one of the first to use backpropagation in

[8]: (1981), *The nobel prize in physiology or medicine 1981*

[10]: Fukushima (1980), *Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position*

[11]: Rumelhart et al. (1986), *Learning representations by back-propagating errors*

[12]: Le Cun et al. (1990), *Handwritten digit recognition with a back-propagation network*

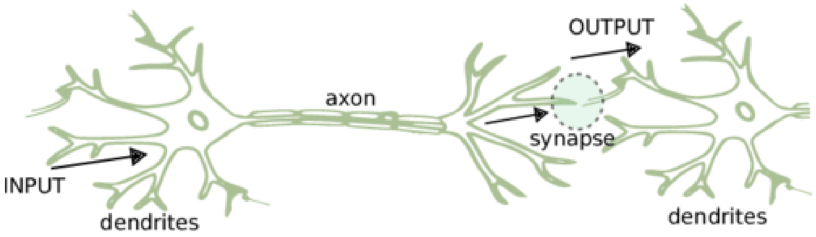


Figure 2.2: Information is passed from one neuron to another. The message enters the dendrites, passes through the axon, and propagates through the synaptic terminal to another neuron. Adapted from Wulfram Gerstner and Werner M. Kistler. *Spiking Neuron Models: Single Neurons, Populations, Plasticity*. Cambridge University Press, 2002 (Ref. [9]).

a visual setting. They combined convolutions and backpropagation into a convolutional neural network to recognize handwritten digits.

2.3 The building blocks of convolutional neural networks

2.3.1 Artificial neural networks

An artificial neural network (ANN) is a computational model inspired by the structure and functioning of the human brain. It consists of connected artificial neurons, also known as nodes or units, organized into layers. Each neuron takes inputs, performs a mathematical operation on them, and produces an output.

ANNs typically have an input layer, one or more hidden layers, and an output layer. Information flows through the network from the input layer, via the hidden layers, to the output layer. The hidden layers contain neurons that transform the input data into a more useful representation. The transformations are dictated by weight matrices W.

Neural networks are designed to learn from data through a process called training. During training, the network adjusts the strengths of connections between neurons, known as weights, based on the patterns in the input data. This process allows the network to recognize and generalize from examples, making it capable of solving complex problems and making predictions.

Optimizing ANNs often relies on backpropagation (from backward propagation of errors) [11]. Mathematically, an ANN g with L layers and activation functions f^l can be described as

[11]: Rumelhart et al. (1986), *Learning representations by back-propagating errors*

$$\hat{y} = g(x) = f^L \left\{ W^L f^{L-1} \left[W^{L-1} \cdots f^1 \left(W^1 x \right) \cdots \right] \right\}, \qquad (2.1)$$

where $\hat{y}$ is the output, given input x. To quantify the error of the model, the loss can be calculated with an appropriate error function $E(y, \hat{y})$, where y is the target corresponding to input x. Calculating $\partial E / \partial w_{ij}$ allows updating individual weights of the network with *e.g.*

$$\Delta w_{ij} = -\eta \frac{\partial E}{\partial w_{ij}}, \qquad (2.2)$$

where η is the learning rate. This optimization algorithm is called stochastic gradient descent (SGD). Backpropagation and SGD form the basis of neural network optimization, but there are other optimization algorithms available such as Adam [13].

[13]: Kingma et al. (2015), *Adam: A Method for Stochastic Optimization*

2.3.2 Convolutional layers

To distinguish a neural network from a convolutional neural network (CNN), at least one layer must be a convolution. A convolution is an operation where a kernel with width w and height h is moved along an input to generate an output. The output, or output feature map, in two dimensions, is

$$\mathbf{o}_{i,j} = \mathbf{b}_{i,j} + \sum_{c=0}^{C-1} (\mathbf{x_c} \circledast \mathbf{u}_c)_{i,j} = \mathbf{b}_{i,j} + \sum_{c=0}^{C-1} \sum_{n=0}^{w-1} \sum_{m=0}^{h-1} \mathbf{x}_{c,n+i,m+j} \mathbf{u}_{c,n,m}, \tag{2.3}$$

where $\mathbf{x}$ is the input possibly containing C multiple channels. The bias $\mathbf{b}$ and weights $\mathbf{u}$ are learnable parameters.

Convolutions can be modified in a few ways that are important for deep learning. The first modification is padding, and specifies the size of an added frame around the input. The frame can have any value, but generally, it is filled with zeros. A second modification is stride, which specifies the step size with which the kernel moves across the input.

A two-dimensional numerical convolution operation with padding and strides is visualized in Figure 2.3. The kernel with size $k = 3$ moves across the input of size $i = 5$ with stride $s = 2$ in both directions.

Convolutions have the useful property that they are equivariant to translations. The equivariance of convolutions to translations implies that learned weights and biases belonging to a convolution can be reused for identifying similar features anywhere in inputs. Any operator that is not equivariant to convolutions may be used as a way to augment data, as the kernel perceives the transformed data as different. Examples of such operators are scaling, rotating, and flipping. Applying these operators on input data trains the model to be invariant to the operator, meaning transformed versions of structures can be identified.

2.3.3 Pooling

Pooling is a form of nonlinear downsampling. To achieve this, typically, a convolution kernel is moved over the input with a stride as big as the kernel itself. This ensures that the downsampling considers measures of input subregions.

Pooling is equivariant to any permutation under the kernel. This results in invariance to local translations. In deep learning, pooling is therefore used to quantify the presence of a pattern, as opposed to finding its position.

Max pooling

The most common form of pooling is max pooling. The kernel finds the maximum value in sub-regions and maps these maximum values per sub-region to a new image. The output

$$\mathbf{o}_{c,i,j} = \max_{n<h,m<w} \mathbf{x}_{c,si+n,rj+m}, \tag{2.4}$$

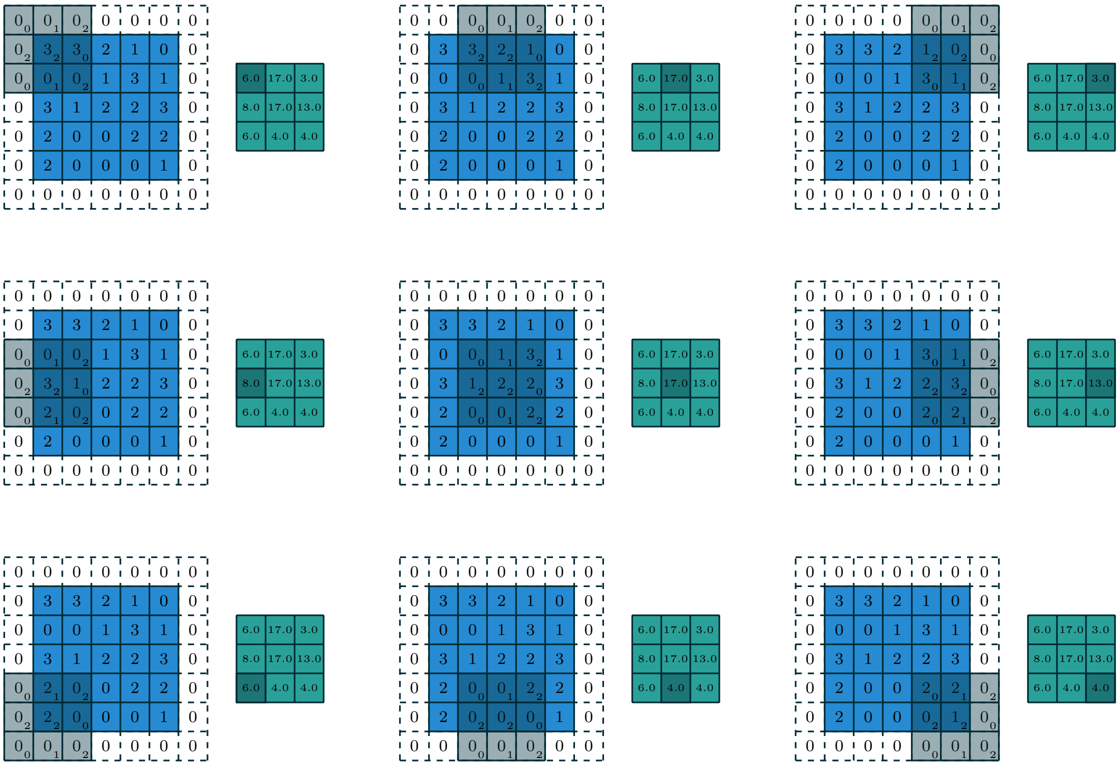


Figure 2.3: Computing the output values of a discrete convolution for two dimension, $i_1 = i_2 = 5$, $k_1 = k_2 = 3$, $s_1 = s_2 = 2$, and $p_1 = p_2 = 1$. Reproduced from Vincent Dumoulin and Francesco Visin. *A guide to convolution arithmetic for deep learning*. 2016 (Ref. [14]).

where rw and sh are the width and height of the input.

Average pooling

Another popular form of pooling is the average pooling, which finds the mean value in sub-regions. The output

$$\mathbf{o}_{c,i,j} = \frac{1}{wh} \sum_{n=0}^{w-1} \sum_{m=0}^{h-1} \mathbf{x}_{c,si+n,rj+m}. \tag{2.5}$$

2.3.4 Activation functions

Saturating activation functions

Neural networks require differential activation functions for backpropagation to update model weights. Sigmoids,

$$\sigma(x) = \frac{e^x}{e^x + 1} = 1 - \sigma(-x), \tag{2.6}$$

are such differentiable functions and often used.

Over the years, neural networks have become deeper, *i.e.* more layers are being added, to make function approximators generalize better. Sigmoids and other saturating curves like the hyperbolic tangent can produce small gradients at either side. For deep networks, gradients resulting from the product of multiple small activation can become too small, preventing the model to learn. This phenomenon is called the vanishing gradient problem.

Rectified linear unit

To overcome the vanishing gradient problem, less saturating activation functions can be used. One such function is the rectified linear unit (ReLU). It is defined as

$$f(x) = x^+ = \max(0, x), \tag{2.7}$$

such that only the positive arguments keep their value. ReLU only saturates on the left side.

Last layer activations

Classification In classification tasks, the last layer usually contains multiple neurons where each neuron stands for a specific classification. To estimate probabilities summing to one, the activation function of the last layer can be replaced by the softmax function,

$$\text{softmax}_i = \frac{e^{z_i}}{\sum_{j=1}^{C} e^{z_i}}, \tag{2.8}$$

where $\mathbf{z} = \mathbf{Wh} + \mathbf{b}$, C the number of classes and i the class of interest.

Regression For regression models, no activation function should be used as the output should be unrestricted.

2.3.5 Loss functions

Neural networks are often updated with a gradient-based optimizer such as SGD. SGD computes gradients of weights with respect to a loss function. Loss functions should be chosen depending on the target. Targets often fall in two categories: regression and classification.

Regression

Mean absolute error One of the most straightforward techniques of calculating the loss is the mean absolute error (MAE). It measures the average difference between every prediction and target, like

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n} |y_i - y_i'|, \tag{2.9}$$

where n is the number of targets per sample, y the prediction and y' the target.

The MAE loss is forgiving, *i.e.*, outliers are weighted as much as predictions close to the target. In training a neural network, focusing on outliers can be beneficial, as those are the cases that the model has difficulty with.

Mean square error To overcome the forgiving nature of the MAE loss, the mean square error (MSE) can be used. It measures the average squared difference between every prediction and target, like

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - y_i')^2. \tag{2.10}$$

Focal MSE To give even more focus on the hard targets, giving them more importance than easy targets can be done through the focal MSE loss (FL) [15]. To give less importance to the easier targets, FL follows

[15]: Lu et al. (2022), *Deep Object Tracking With Shrinkage Loss*

$$FL = \left(\frac{2}{1 + e^{-\beta|y-y'|}} - 1\right)^{\gamma}(y_i - y_i')^2, \tag{2.11}$$

where increasing γ increases the number of targets regarded as easy and β regulates the speed with which the first part of the curve increases.

Classification

For classification, the last layer gives an estimate of class probabilities. Targets are often binary: probability of zero and one for the negative and positive class, respectively.

Cross entropy Cross entropy can be used to calculate the loss between probabilities and their targets. It is defined as

$$CE(x,y) = \frac{1}{N}\sum_{n=1}^{N}\left[-\sum_{c=1}^{C}\log\frac{e^{x}_{n,c}}{\sum_{i=1}^{C}e^{x_{n,i}}}y_{n,c}\right], \tag{2.12}$$

where x is the probability output, y the target, C the number of classes, and N the number of batches used.

Binary cross entropy In the specific case where there are only two classes, Equation 2.12 can be reduced to

$$BCE(x,y) = \frac{1}{N}\sum_{n=1}^{N}[y_n \cdot \log x_n + (1-y_n)\cdot\log(1-x_n)]. \tag{2.13}$$

2.4 Training a neural network

2.4.1 Training

At the start of training, a neural network has its weights and biases initialized. Most probably, the model is not capable of mapping input to output in a robust manner. To achieve this, repeatedly using backpropagation to update the model parameters aims to shape the model in the direction of minimizing the loss between target and model output. The neural network is presented with the input data in batches. Every batch, the model is updated with the backpropagation algorithm. One cycle of using all the batches is called an epoch. A training consists of multiple epochs.

For every $\mathscr{B}$th batch, a loss $\mathscr{L}_{\mathscr{B}}$ can be defined that is used by backpropagation to penalize the model performance. Taking the average of all batch losses is the epoch loss,

$$\mathscr{L}_{\text{epoch}} = \frac{1}{N_{\mathscr{B}}} \sum_{i=0}^{N_{\mathscr{B}}} \mathscr{L}_{\mathscr{B}}, \tag{2.14}$$

Tracking $\mathscr{L}_{\text{epoch}}$ shows how quickly the model is learning.

To see if the model generalizes, it is standard practice to have a hold-out set, that the model does not learn from, but only uses to calculate the validation loss. Ideally, this validation loss follows a similar trajectory as the training loss. If the validation loss diverges upwards from the training loss, the model is overfitting: it fails to generalize to unseen, but similar data. To remedy this, there are multiple possible solutions. Solutions include dropout, batch normalization, or deeper and wider networks.

Dropout

Chance of overfitting can be reduced by applying methods of regularization. One regularization method is dropout. It prevents neurons from co-adapting, which would otherwise reduce the chance of the model to perform well on external validation sets [16]. With dropout, individual neurons are activated with probability p, effectively dropping neurons randomly.

[16]: Srivastava et al. (2014), *Dropout: A Simple Way to Prevent Neural Networks from Overfitting*

Batch normalization

Batch normalization (BN) [17] is a technique to shift and scale batches akin to standardization. It can be implemented as a layer in any neural network. Per batch and per dimension, the mean and standard deviation of the input are calculated. Then, the input is standardized with

$$\hat{x}_i = \frac{x_i - \mu_{\mathscr{B}}}{\sqrt{\sigma_{\mathscr{B}}^2 + \epsilon}}, \tag{2.15}$$

where $\mu_{\mathscr{B}}$ and $\sigma_{\mathscr{B}}$ are the mean and unbiased standard deviation of the batch, and ϵ is a small number for numerical stability when the variance is small. The standardized input is then mapped through

$$y_i = \gamma \hat{x}_i + \beta, \tag{2.16}$$

where γ and β are learnable parameters learned in a sub-network.

When batch normalization is applied after a convolutional layer, the bias term of the convolution becomes redundant and can be set to zero to avoid unnecessary operations.

[17]: Ioffe et al. (2015), *Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift*

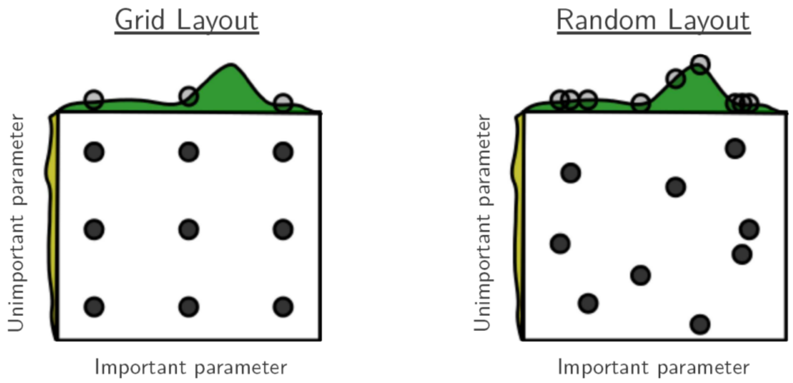


Figure 2.4: Grid and random search of nine trials for optimizing a function $f(x,y) = g(x) + h(y) \approx g(x)$ with low effective dimensionality. Above each square $g(x)$ is shown in green, and left of each square $h(y)$ is shown in yellow. Reproduced from James Bergstra and Yoshua Bengio. 'Random Search for Hyper-Parameter Optimization'. In: *J. Mach. Learn. Res.* 13 (2012) (Ref. [19]).

BN has been shown to have a regularizing effect [18], although combining it with dropout is disputed. More often than not, using both BN and dropout leads to worse results on the test set.

[18]: Bjorck et al. (2018), *Understanding Batch Normalization*

2.4.2 Hyperparameter optimization

A machine learning model uses training data to learn parameters to map input to output data best. However, there are parameters that cannot be learned, but greatly influence the training outcome. These parameters are hyperparameters. Examples of hyperparameters are the batch size, learning rate, learning rate scheduler and its parameters, optimizer algorithm, etc. These parameters span a configuration space $\mathscr{C}$. Parameters can be categorical (type of optimizer, learning rate scheduler, etc.) or integers (batch size, number of iterations, etc.), or continuous decimals (learning rate, weight decay, etc.). Ideally, parameters are sampled exhaustively. This way, the best possible set of parameters can be found. However, this can be computationally expensive. Moreover, when using a continuous variable, it is no long possible to exhaustively sample parameters. To engage this problem, various algorithms have been developed to sample hyperparameters from the high-dimensional distribution.

Grid search and random search

The most straightforward technique of finding the best set of hyperparameters is grid search. With grid search, parameters are sampled exhaustively using equidistant spacing in each dimension.

A drawback of grid search is that optima can reside outside the hyperparameter set that grid search produces. Random search [19] aims to find optima in the gaps using random search. With the same number of trials, random search has a higher probability for trials to find the global optimum. This is because trials explore the whole distribution as opposed to just a few points in individual dimensions. Figure 2.4 shows the differences between grid search and random search and advocates the use of the latter.

[19]: Bergstra et al. (2012), *Random Search for Hyper-Parameter Optimization*

Tree Parzen estimator

Random search requires trials in regions that are unpromising which is inefficient. A tree-structured Parzen estimator (TPE) [20] approach aims to model the probability of a hyperparameter[1], given a loss value. That probability consists of two distributions, describing the good and bad values:

$$p(c|L) = \begin{cases} p(c|L > L^*) = p(c|\text{bad}) \\ p(c|L \leq L^*) = p(c|\text{good}), \end{cases} \tag{2.17}$$

where c is drawn from configuration space $\mathscr{C}$ and L is the loss. L^* a loss above which losses are considered bad. TPE chooses L^* to be a fraction of observed L values, such that $p(\text{good}) = \gamma$. A promising candidate has low probability under $p(c|\text{bad})$ and high probability under $p(c|\text{good})$. Therefore, c is promising if

$$\text{promisingness}(c) \propto p(c|\text{good})/p(c|\text{bad}) \tag{2.18}$$

is high. Ref. [20] shows that this ratio is proportional to the expected improvement [22]. The configuration responsible for the maximum of promisingness(c) is used as the next trial. Results of that trial are now categorized as good or bad, and the iterative process continues.

[20]: Bergstra et al. (2011), *Algorithms for Hyper-Parameter Optimization*

[22]: Jones (2001), *A Taxonomy of Global Optimization Methods Based on Response Surfaces*

Successive Halving and Hyperband

Although $\mathscr{C}$ can be sampled more efficiently with TPE, trials still use the full computational budget, even if it is apparent that the trial is unpromising early on. Early terminating (or pruning) these underperforming trials speeds up hyperparameter optimization. Pruning trials can be done using Successive Halving (SH) [23]. Given a computational budget B, *e.g.* number of epochs, the number of trials T, and the halving rate γ, SH performs $\log_\gamma(T)$ rounds. The budget is distributed uniformly over the trials. Every round, 100 % × $1/\gamma$ of the trials are discarded based on their performance. Surviving trials are allowed twice the budget and are again discarded when they have used up their budget. This iterative process continues until one trial remains.

[23]: Jamieson et al. (2016), *Non-stochastic Best Arm Identification and Hyperparameter Optimization*

When using SH, two variables need to be considered, and possible manually tuned. The more available budget, pruning decisions are made more confident. Higher halving rates lead to more and more aggressive pruning with the risk of pruning good candidates early.

There is a trade-off between T and B. Suppose T is large, then each trial gets a small amount of budget, but many configurations are explored. Conversely, if T is small, then each trial gets much budget, at the cost of exploring the number of configurations. This T/B trade-off is addressed by Hyperband (HB) [24] by performing a grid search over feasible values of T. HB invokes SH multiple times. Every invocation of SH is called a bracket. In the end, HB returns the best configuration possible just like SH, but diminishing the dependence on manually choosing a good T.

[24]: Li et al. (2017), *Hyperband: A Novel Bandit-Based Approach to Hyperparameter Optimization*

[20]: Bergstra et al. (2011), *Algorithms for Hyper-Parameter Optimization*

[1] Or a set of hyperparameters in the case of multivariate TPE [21]

Parameter importances

Not every hyperparameter is as important as others. Hutter, Hoos, and Leyton-Brown [25] describe the fANOVA algorithm to quantitatively assess the importance of every hyperparameter. Knowing the importance of a variable gives more insight into interactions and relative importance between hyperparameters.

[25]: Hutter et al. (2014), *An Efficient Approach for Assessing Hyperparameter Importance*

2.5 Image quality

A convolutional neural network receives a stream of input images with varying quality. For example, microscopy images from deep inside tissue are presumably noisier and/or less bright than images taken near the surface. Neural networks have trouble learning from bad images, as they lack structures that trigger neurons to output predictions close to targets. Excluding noisy images might increase performance [26]. Koho et al. [27] suggest some measures to quantify image quality. Here, the entropy and kurtosis are discussed.

[26]: Blokker et al. (2022), *Fast intraoperative histology-based diagnosis of gliomas with third harmonic generation microscopy and deep learning*

[27]: Koho et al. (2016), *Image Quality Ranking Method for Microscopy*

2.5.1 Shannon entropy

The quality of an image may be described by the amount of information that is contained within it. The information can be quantified by how surprising it is to contain specific content. For instance, knowledge that a rare event will occur has high informational value, while knowledge that a probable event will happen has low informational value. Given a random variable X, the information, or entropy, is defined as

$$H = \mathbb{E}[-\log p(X)] = -\sum p(x) \log p(x), \tag{2.19}$$

where $\mathbb{E}[\dots]$ is the expectation operator, and $p(x)$ is the probability of x occurring. The logarithm satisfies the boundary condition that an event is not surprising if its probability of occurring is one.

For images, Equation 2.19 can be rewritten as

$$H_I = -\sum_{i}^{n} P_i \log_2 P_i, \tag{2.20}$$

where P_i is the normalized image histogram at bin index i [27]. The base of the logarithm is chosen to be two, such that the entropy is in units of bits.

[27]: Koho et al. (2016), *Image Quality Ranking Method for Microscopy*

For images having many different intensities, the entropy is high, because of the knowledge that pixel intensities having lower probability. The intuition for images with high entropy tending to carry more information is the same as taking images with longer exposure times: the number of pixels with a certain intensity will increase. This may be most apparent in dark regions, which would benefit from more illumination.

2.5.2 Kurtosis

Another measure for image quality is kurtosis of the power spectrum. Kurtosis measures the outliers of a distribution and is given by

$$\kappa = \frac{\mu_4}{\sigma^4}, \tag{2.21}$$

where σ is the standard deviation and μ_4 is the fourth moment about the mean. The nth moment about the mean is defined as

$$\mu_n = \mathbb{E}[(X - \mathbb{E}[X])^n] = \int_{-\infty}^{\infty} (x - \mu)^n p(x)\,\mathrm{d}x. \tag{2.22}$$

Distributions having a kurtosis of zero are mesokurtic, meaning they resemble a normal distribution. Posivite kurtosis means that the distribution is leptokurtic. A leptokurtic distribution has tails with more weight compared to the normal distribution, such as the Poisson or Laplace distribution. Negative kurtosis means that the distribution is platykurtic. Platykurtic distributions have thinner tails, such as the Bernoulli distribution.

Kurtosis can be calculated on the upper part of the power spectrum of an image [26, 27]. If the upper part of the power spectrum is very leptokurtic compared to other images in the dataset, it may indicate that the image is an outlier and is significantly different from the mean.

[26]: Blokker et al. (2022), *Fast intraoperative histology-based diagnosis of gliomas with third harmonic generation microscopy and deep learning*
[27]: Koho et al. (2016), *Image Quality Ranking Method for Microscopy*

2.6 Explainable AI

A significant number of users of a trained AI generally view the model as a black box that simply maps input to output. How this box is constructed and why it results in a particular outcome is often overlooked. Meanwhile, techniques to give insight into the black box (explainable AI, XAI) have been developed. These techniques fall in roughly three categories: gradient and perturbation based methods and explainable models. Gradient based methods rely on gradients calculated during the backward pass and use these to find which parts of the input contribute to the output most. Perturbation methods perturb the input to see how the output changes. Large output changes yield large attributions. Explainable models output some form of attention maps as intermediate steps to their prediction.

Explainability is particularly important in clinical settings where merely relying on AI output is sometimes unethical. XAI gives users and patients more confidence in the prediction so that specialists can proceed with treatment.

2.6.1 Occlusion

Occlusion [28] is an XAI pertubation technique. The method replaces patches of the input with a baseline value. *e.g.* for images, patches can consist of any shape and the baseline value can be 0, practically making a patch black and removing all information at the patch's location and the connections with neighboring pixels. In the original paper, occlusion

[28]: Zeiler et al. (2014), *Visualizing and Understanding Convolutional Networks*

is used to systematically cover parts of foreground objects, to gain confidence in the AI using foreground objects to predict the output. If the AI still recognizes an object from an image where the object has been cut out, the AI may use background for its prediction.

A generalized occlusion algorithm includes a moving patch. The patch, mostly a rectangle of a given size and value, is placed on the image. The AI computes an output, given the masked input. The masked output is subtracted from the original output. This difference divided by the patch size is assigned to the patch. A new patch is placed, and the iterative process continues until the patches have been placed on all possible input locations.

2.7 Comparing model performance

When testing and comparing models, it is important to discern *outperform* from *performing comparably*. To that end, a null and alternative hypothesis (H_0 and H_a, respectively) have to be formulated. Before testing, a confidence level α such as 0.05 or 0.01 has to be chosen. Then, p is calculated, which is

$$p = P(\text{reject } H_0 \text{ and suggest } H_a | H_0 \text{ is true}). \tag{2.23}$$

If $p < \alpha$, H_0 may be rejected, suggesting H_a is true.[2]

In practice, p is calculated via a test statistic. The test statistic might follow *e.g.* a Student's t-distribution or a normal distribution. The distribution has to reflect the assumed distribution of the sample. To compare two means from different samples with unequal variances, Welch's t-test can be used. The test statistic

$$t = \frac{\bar{\mathbf{X}}_A - \bar{\mathbf{X}}_B}{\sqrt{\sigma^2_{\bar{\mathbf{X}}_A}/(n_A - 1) + \sigma^2_{\bar{\mathbf{X}}_B}/(n_B - 1)}}, \tag{2.24}$$

where $\bar{\mathbf{X}}_A$ and $\sigma_{\bar{\mathbf{X}}_i}$ are the ith sample mean and standard deviation. n_i is the number of samples in X_i. Using t and a two-sided test,

$$\begin{aligned} p &= P(|t| \geq t) && (2.25) \\ &= 1 - F_{|t|}(t), && (2.26) \end{aligned}$$

where F is the cumulative distribution function.

Example 2.7.1 Two models A and B have been tested on three separate datasets (commonly done with cross-validation). Beforehand, a significance level of $p = 0.05$ is chosen. The models have accuracies

$$\begin{aligned} \mathrm{acc}_A &= \{0.5, 0.7, 0.9\}, && (2.27) \\ \mathrm{acc}_B &= \{0.5, 0.6, 0.7\}. && (2.28) \end{aligned}$$

The null hypothesis $H_0 : \mu_A = \mu_B$, *i.e.* the mean of the underlying accuracy distributions are equal. The alternative hypothesis $H_a : \mu_A \neq \mu_B$. Welch's

[2] p or the p-value is often misinterpreted with the reverse condition: $p = P(H_0 \text{ is true}|\text{sample statistics})$.

t-test for two samples with unequal variance gives

$$t \approx \frac{0.7 - 0.6}{\sqrt{0.16^2/2 + 0.082^2/2}} \approx 0.77. \tag{2.29}$$

Using the inverse Student's t cumulative distribution function, $p = 0.50$. If $p < 0.05$, the null hypothesis could have been rejected, suggesting the alternative hypothesis. But $p > 0.05$, and therefore the null hypothesis cannot be rejected, meaning both models perform comparably with a significance level of $\alpha = 0.05$.

3

Developing and validating a strain-stress regression model on second harmonic generation images from old adult skin tissue

Abstract

Background and objective Second harmonic generation (SHG) microscopy allows for imaging of biological tissue at micrometer scale such as collagen fibers. Mechanical human skin stretch experiments were done to relate HHG images to stretch properties. Stretch properties might be extracted by AI models to substitute mechanical measurements. A skin stretch regression model (Skinstression) is developed and validated to compute the stress-strain curves corresponding to SHG images of human skin tissue.

Methods A holdout study was conducted on SHG data of human skin tissue. Outcomes of interest were the maximum stress, strain offset and maximum Young's modulus which together construct a stress-strain logistic curve. A convolutional neural network was developed and validated. The performance of the models was assessed by the coefficient of determination R^2 and occlusion was used to possibly explain predictions. Artificially adding and removing collagen was done to attack the model.

Results SHG skin tissue images of 18 old adult (5 men, 4 women, 6 unknown, ages 61 yr to 94 yr) were used. The model achieved a mean R^2 of -0.36 (SE 0.60) on the test set. Occlusion did not give insight into stretch property predictions. Adversarial attacks seem to induce predictions corresponding with adding or removing collagen.

Discussion Skinstression needs further training and external validation. After additional training and validation, the updated model may be used to replace mechanical skin stretch measurements and ultimately analyze live microendoscope images to be used by plastic surgeons.

3.1 Introduction

Human skin tissue is built by a series of layers, one of which is the dermis layer. The dermis contains a network of collagen fibrils, dictating the stretch of skin tissue. Measuring the strain-stress response of skin tissue gives insight into the skin's strength and elasticity that protects the body from external forces. A recent study aims to show the connection between collagen density and stretch [29]. To measure the strain-stress response of skin tissue, mechanical measurements have to be performed.

Second harmonic generation (SHG) imaging allows imaging collagen and two-photon excitation microscopy (2PEF) elastin. Setups have been built to collect SHG and 2PEF signals simultaneously, allowing for rich skin tissue imaging. Collagen fibers are clearly visible. SHG images of the collagen networks in conjunction with the strain-stress curves suggest that the images already contain stretch information. Retrieving complex features from labelled images can be done using supervised deep learning. Supervised deep learning is considered a black-box technique that aims to learn a mapping from input to output. With the SHG images and corresponding stress-strain measurements at hand, Skinstression (from skin stretch regression) is developed with the ultimate goal to replace mechanical measurements on skin tissue to quantify skin stretch. Possibly, the model can be used to non-invasively investigate physical parameters to aid plastic surgery. For example, the flexibility of skin tissue prior to excision determines the amount of manual continuous or cyclic stretching needed to close a gap after excision [30].

Efforts to develop such a model have already been made by Soylu [31]. However, those methods do not consider complete separation of training and test sets in both inference and label creation, possibly leading to biased results. Moreover, only one slice of larger image stacks have been used. The original model does not incorporate physical properties of the strain-stress curves but relies on principal component analysis.

Frequently, machine learning models and neural networks in particular lack the ability to explain how the model comes to its conclusions. Meanwhile, explainable artificial intelligence (XAI) techniques exist to shed light on the inner workings of algorithms, but are used sparingly. In the context of skin tissue, XAI could give insight into where the strength and elasticity comes from. A human expert might recognize straight collagen fibrils as a stiff network [32], but it is interesting to see if an AI explains stiffness in the same manner.

The objective of this study is to extend Soylu's model. To this end, an application, Skinstression, will be developed and validated with separate training and validation data. A new, physics informed neural network will be implemented for explainability. The data will not only consist of single slices from depth scans, but of subsets considering multiple slices per depth scan. Moreover, XAI procedures will be adopted to better explain the black box output.

The purpose of the product is to accompany or possibly replace mechanical strain-stress measurements on skin tissue. Explainability techniques might shed light on how top level collagen structures provide strength and elasticity to skin. Unintentionally, the project may be used more generally to train other convolutional neural networks for regression.

[29]: Zhou et al. (n.d.), *Three-dimensional Characterization of Mechanical Properties and Microstructures of Human Dermal Skin*

[30]: Verhaegen et al. (2012), *Adaptation of the dermal collagen structure of human skin and scar tissue in response to stretch: An experimental study*

[31]: Soylu (2022), *Developing a non-invasive approach to estimate physical parameters of skin tissue from HHG microscopy using convolutional neural networks*

[32]: Holzapfel (2001), *Biomechanics of Soft Tissue*

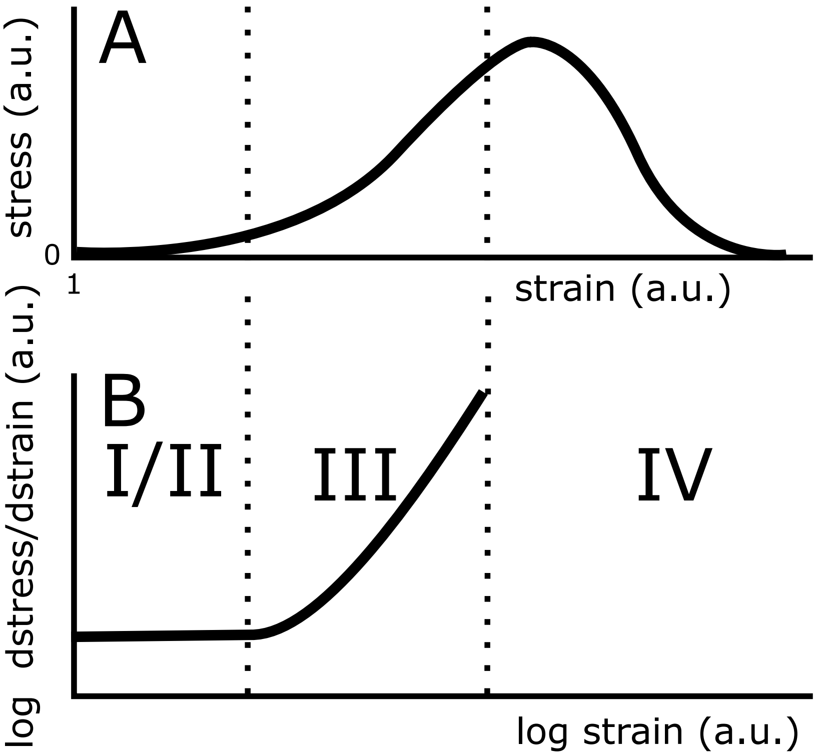


Figure 3.1: Stress-strain curve of human skin tissue. The curve is divided in three regions. I: the toe region. The fiber structure here consists of mostly straight fibers. II: the heel region. Fibers begin to stretch. III: the leg region. Fibers stretch maximally and react with maximum force. IV: break region. Fibers break and give up ability of applying force.

3.2 Theory

3.2.1 Searching for a simple skin strain-stress model

Supervised learning requires targets for the model to train on. Ideally, individual targets allow for physical interpretation and can together describe all the available data.

Empirical strain-stress regions

Although skin tissue has a complex nature, measurements to quantify skin stretch show similar features. Measurements show four domains: the toe, heel, leg and break domain (Figure 3.1A). The toe region is at the very start of the curve. This region is relatively flat as the fiber network consists of mostly straight fibers. Therefore, the fibers cannot exert force as a reaction to external stretching force. However, in the heel region where skin tissue is stretched more, fibers can exert more force. When enough force is exerted on the tissue, fibers stretch maximally and fibers react with maximum force in the leg region. This region is observed to be roughly linear. Overstretching the tissue then breaks the fiber network, decreasing the possibility to exert force.

Exponential

Strain-stress curves can also be visualized by showing the log derivative of stress with respect to strain against the log of strain as in Figure 3.1B. Typically, this figure has three regions. The first region indicates a linear relationship between small forces and small strain. Then, the derivative increases until it reaches a purely exponential part [32]. If skin stretching follows this kind of behavior, a simple mathematical model can be derived. Inspecting the figure, the linear part shows the ordinary differential equation

$$\frac{\mathrm{d}\sigma}{\mathrm{d}\gamma} \propto \sigma, \tag{3.1}$$

where $\gamma = \chi - 1$. Solving for σ, we get

$$\sigma \propto e^{\lambda\gamma}, \tag{3.2}$$

where λ is some factor dictating the speed with which the exponential increases. At no extension, $\gamma = 0$, it can be assumed that there is no stress. Therefore,

$$\sigma \propto e^{\lambda\gamma} - 1. \tag{3.3}$$

At small extensions, $\lambda\gamma \ll 1$, $e^{\lambda\gamma} \approx (1+\lambda\gamma+\ldots)$ using a Taylor expansion. So

$$\sigma_{\lambda\gamma\ll 1} \propto 1 + \lambda\gamma + \ldots - 1 \approx G_0\gamma, \tag{3.4}$$

where G_0 is some linear coefficient at small extensions. In this work, $\gamma = \chi - 1$, where χ is the stretch. The full expression then becomes

$$\sigma = \frac{G_0}{\lambda}\left(e^{\lambda(\chi-1)} - 1\right). \tag{3.5}$$

This model assumes that data follows the previously described curve where there is a small rise at small extensions and an indefinitely exponentially increasing stress for larger extensions.

The exponential model is fit to some stress-strain curves using OriginPro [33].

[32]: Holzapfel (2001), *Biomechanics of Soft Tissue*

[33]: (2022), *OriginPro*

Principal component analysis

In an earlier study [31], principal component analysis (PCA) is used to reduce the dimensionality of the strain-stress data. In summary, after PCA, every measurement Y can be approximated by

$$Y \approx Y_{\mathrm{PCA}} = \mathbf{AV} + \bar{Y}, \tag{3.6}$$

where $\mathbf{A}$ and $\mathbf{V}$ are matrices containing respectively the eigenvalues and eigenvectors of the measurement data. $\bar{Y}$ is the measurement mean. Choosing the first p principal components allows for dimensionality reduction.

Using PCA to create eigenvalues to weight the eigenvectors has some caveats. First, the training and test sets must be treated separately. The test set has to be projected on the space spanned by the first p eigenvectors of the training set. This may induce problems as the test set could contain information that does not come close to Second, PCA depends on interpolation, *i.e.* every strain-stress curve must be formed by either a set of strain or stress values. This reduces the domain of the data.

[31]: Soylu (2022), *Developing a non-invasive approach to estimate physical parameters of skin tissue from HHG microscopy using convolutional neural networks*

Logistic curve The empirical observations where the force response of skin tissue changes states, suggests a logistic curve, which can be written as

$$\sigma = \frac{\sigma_{\text{max}}}{1 + e^{-E_{\text{max}}(\gamma - \gamma_c)}}, \tag{3.7}$$

where σ and γ are the stress and engineering strain, σ_{max} is the maximum stress, E_{max} is the maximum Young's modulus and γ_c the strain offset. This equation assumes that there is a maximum force that the tissue can exert, in contrary to the theoretical approach in Subsection 3.2.1.

Using the logistic curve as an alternative to PCA has two major advantages. Every curve can be treated separately and measurements can contain data across arbitrary domains and with arbitrary intervals as no interpolation is necessary.

3.2.2 Label density smoothing

The targets calculated with logistic curve fitting result in a non-uniform distribution. Data imbalance reduces the ability of a neural network to learn outliers. This may have significant impact on the test results. To deal with imbalanced data, various techniques have been developed. One of those techniques is label density smoothing (LDS) [34]. It is specifically designed for deep neural networks to learn from imbalanced continuous targets.

[34]: Yang et al. (2021), *Delving into Deep Imbalanced Regression*

LDS computes the effective label density distribution,

$$\tilde{p}(y') = \int_Y k(y, y')p(y)dy, \tag{3.8}$$

where $k(y, y')$ is a symmetric kernel, $p(y)$ the number of label y present in the training data and $\tilde{p}(y')$ the effective density of target label y'. Reweighting the loss function with the inverse (square root) of $\tilde{p}(y')$ addresses target imbalance.

3.2.3 Goodness of fit

One possible way to quantify how good a fit is, is to calculate the coefficient of determination. The coefficient of determination of a dataset y and its prediction f, is calculated with

$$R^2 = 1 - \frac{SS_{\text{tot}}}{SS_{\text{res}}}, \tag{3.9}$$

where

$$SS_{\text{tot}} = \sum_i (y_i - \bar{y})^2, \tag{3.10}$$

with $\bar{y}$ the mean of y, and

$$SS_{\text{res}} = \sum_i (y_i - f_i)^2. \tag{3.11}$$

If $R^2 = 1$, the prediction perfectly fits the data. There are some caveats to using R^2 to determine the goodness of fit. Most notably, R^2 does not indicate whether the model is the correct model or whether there are enough data points to draw a conclusion.

3.3 Methods

3.3.1 Data

Sources of data

Human skin tissue was excised from cadavers and healthy subjects for previous studies at the Red Cross Hospital in Beverwijk, the Netherlands [29, 35]. Pieces of these tissues were imaged with multiphoton microscopy and their stress-strain response curves were measured mechanically, see Figure 3.2 [29, 35]. Data is acquired in batches from April 2021 until July 2022. Development and testing data come from the same source.

[29]: Zhou et al. (n.d.), *Three-dimensional Characterization of Mechanical Properties and Microstructures of Human Dermal Skin*
[35]: Haasterecht et al. (2023), *Visualizing dynamic Three-dimensional changes of human reticular dermal collagen under mechanical strain*
[29]: Zhou et al. (n.d.), *Three-dimensional Characterization of Mechanical Properties and Microstructures of Human Dermal Skin*
[35]: Haasterecht et al. (2023), *Visualizing dynamic Three-dimensional changes of human reticular dermal collagen under mechanical strain*

It is unknown if individuals received treatment relevant for this study.

The sample size is arrived at taking into account all previously included subjects and excluding abdomen data and scar tissue. This amounts to a total of 1649 SHG images from 63 samples to train on. Due to the limited amount of participants, individuals with unknown gender or age were included.

Data preprocessing

Depth stack images with a size of 1000 × 1000 with a planar resolution of 1 µm of all skin tissues were kindly provided by M. Zhou. All stacks were separated into slices.

Images consist of three channels: third and second harmonic generation, and autofluorescence. The SHG channel is chosen as it is assumed to only contain collagen information.

The SHG images are enhanced with contrast limited adaptive histogram equalization (CLAHE) [36] using scikit-image [37] to equalize importance of dark and bright regions.

The enhanced images are then transformed with a Yeo-Johnson transform using Scipy [38] such that the histogram of all images is as normal as possible. Normalization generally accelerates training [39].

[36]: Zuiderveld (1994), *Contrast Limited Adaptive Histogram Equalization*
[37]: Walt et al. (2014), *scikit-image: image processing in Python*
[38]: Virtanen et al. (2020), *SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python*
[39]: Huang et al. (2023), *Normalization Techniques in Training DNNs: Methodology, Analysis and Application*

The transformed images are standardized by subtracting the total mean and total standard deviation of the complete transformed image set, like

$$X_{\text{out}} = \frac{X_{\text{in}} - \mu}{\sigma}, \tag{3.12}$$

where

$$\mu = \frac{1}{N} \sum_i \sum_j \sum_k X_{i,j,k}, \tag{3.13}$$

and

$$\sigma = \sqrt{\frac{1}{N} \sum_i \sum_j \sum_k \left(X_{i,j,k} - \mu\right)^2}, \tag{3.14}$$

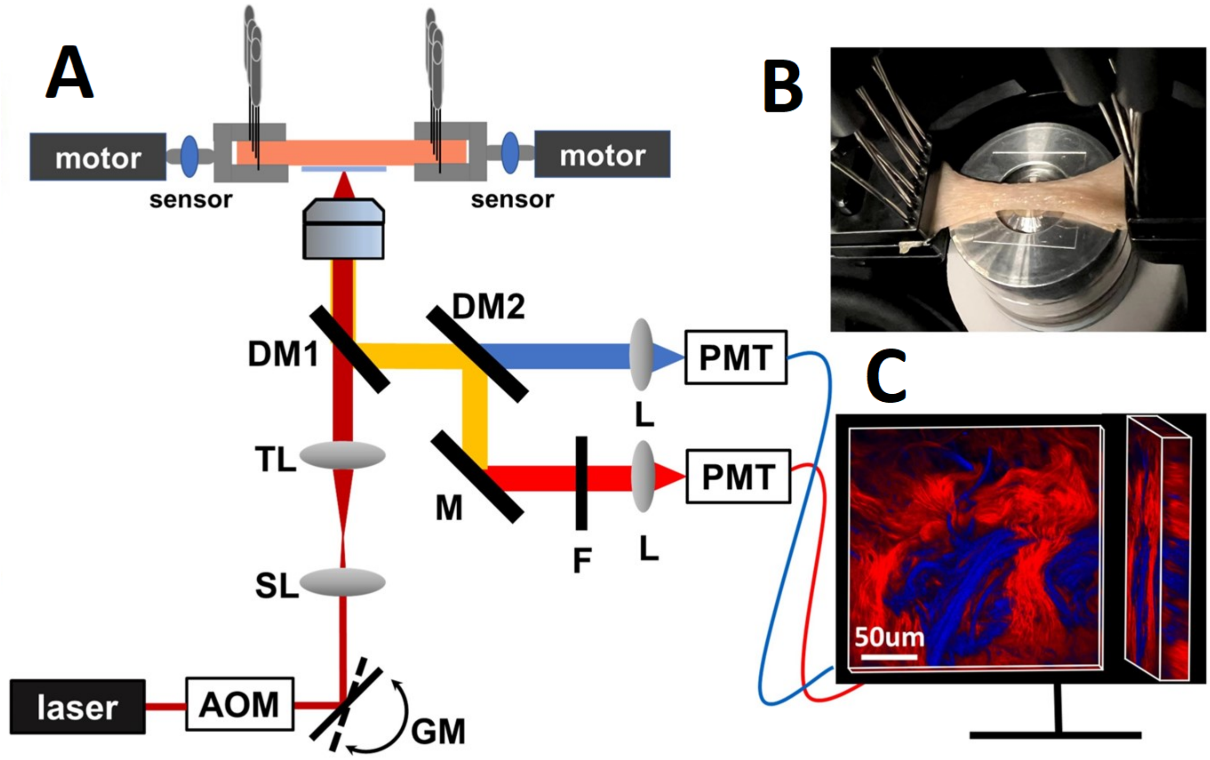


Figure 3.2: Skin stretch setup. The schematic of the experimental setup (A) shows a femtosecond pulse laser, which central wavelength is 1050 nm with pulse duration less than 80 fs; AOM- acousto-optic modulator; SL-scan lens; TL-tube lens, focus tunable; DMP1-dichroic mirror reflecting backscattering signals from fundamental photons; DMP2 dichroic mirror splitting 2PEF and SHG channels; M- Mirror; F- bandpass filter, F520/35; L- focusing lens; PMT-photomultiplier tube detectors. The PMT signals are shown as image stacks (C) where red color represents collagen fibers and blue color represents elastin fibers. A photograph of the skin stretching for 150 % is shown (B). Adapted with permission from Mengyao Zhou, J. González Patrick, Ludo van Haasterecht, Alperen Soylu, Maria Mihailovski, Paul van Zuijlen, and Marie Louise Groot. 'Three-dimensional Characterization of Mechanical Properties and Microstructures of Human Dermal Skin' (Ref. [29]).

with N the total number of pixels, k an individual image and i, j the pixel in the horizontal and vertical dimension, respectively.

The images are resized to 258 × 258 to fit into the neural network.

Image selection

SHG microscopy images from skin tissue suffer from optical phenomena. The most evident problem is that imaging deeper into the tissue, photons are detected with less spatial accuracy because of scattering. The deeper photons travel into tissue, the more possible paths photons can take to return to the detector. Moreover, the chance of photons getting absorbed by the tissue increases with penetration depth. Therefore, less photons get reflected from deeper tissue.

To counter these optical effects, inspired by Koho et al. [27] and Blokker et al. [26], measures to quantify image quality can be obtained. With this, images can be sorted to this measure and the top k images with best quality can be used to train the network, thus excluding noise.

Candidates for this measure are Shannon entropy, kurtosis, and skew for reasons explained in 2.5 These quality measures are calculated per image using PyImageQualityRanking [27], such that the quality measure can be validated by observing manually.

[27]: Koho et al. (2016), *Image Quality Ranking Method for Microscopy*

[26]: Blokker et al. (2022), *Fast intraoperative histology-based diagnosis of gliomas with third harmonic generation microscopy and deep learning*

[27]: Koho et al. (2016), *Image Quality Ranking Method for Microscopy*

Data augmentation

To make the model more robust, data augmentation is applied. Before resizing, the preprocessed images are cropped randomly from 1000×1000 to 700×700 preserving the aspect ratio. The global brightness is adjusted randomly with ±30 %. The images are then randomly mirrored horizontally and vertically with a probability of 50 %. All data augmentations were performed with Torchvision [40].

[40]: TorchVision maintainers and contributors (2016), *TorchVision: PyTorch's Computer Vision library*

3.3.2 Model

The outcome of interest of the model are strain-stress response curves from SHG images from individual skin tissue pieces. The model is named Skinstression (from skin stretch regression).

This section shows the methods obtain a trained model from stress-strain curves and images to use it for inference and interrogate it with XAI techniques. Figure 3.3 shows the model development flow.

Predictor pre-selection

As discussed in Subsection 3.2.1, there are three predictor candidates. These candidates are tested against the original strain-stress curves.

Strain-stress curves for all individuals were kindly provided by M. Zhou. The curves only include points where the skin extension is larger than zero and the force positive.

Exponential and logistic curve The exponential and logistic models are fitted to all raw strain-stress curves with Scipy [38]. The optimal parameters were used as targets for training the model. The goodness of fit is determined by the coefficient of determination (Subsection 3.2.3). A fit is considered good if $R^2 \approx 1$ and it passes reasonably through all data points. In particular, the exponential regime of the fit should describe the leg part of the curve.

[38]: Virtanen et al. (2020), *SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python*

Principal component analysis PCA requires curves to align on at least one axis. The first step to achieve this is excluding all stretch values above the stretch of the maximum of the shortest curve. Soylu [31] did linear interpolation on the curves and restricted both stretch and stress to minim peak value. PCA on two variables requires only one shared set of points. Moreover, results of Soylu show knicks in the PCA reconstructions near the end of the curves, which could originate from a limited amount of data or linear interpolation. Therefore, in this study, a non-uniform, univariate, interpolating spline was fitted to all points using Scipy [38] and the stress was calculated from the spline at the stetch values of the curve with the lowest maximum stretch. After PCA on the complete dataset using Scikit-learn [41], the explained variance per component was calculated and used as a method to find an appropriate number of principal components. From these principal components, the curves where reconstructed using Equation 3.6. The goodness of fit is determined by the coefficient of determination (Subsection 3.2.3) and

[31]: Soylu (2022), *Developing a non-invasive approach to estimate physical parameters of skin tissue from HHG microscopy using convolutional neural networks*

[38]: Virtanen et al. (2020), *SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python*

[41]: Pedregosa et al. (2011), *Scikit-learn: Machine Learning in Python*

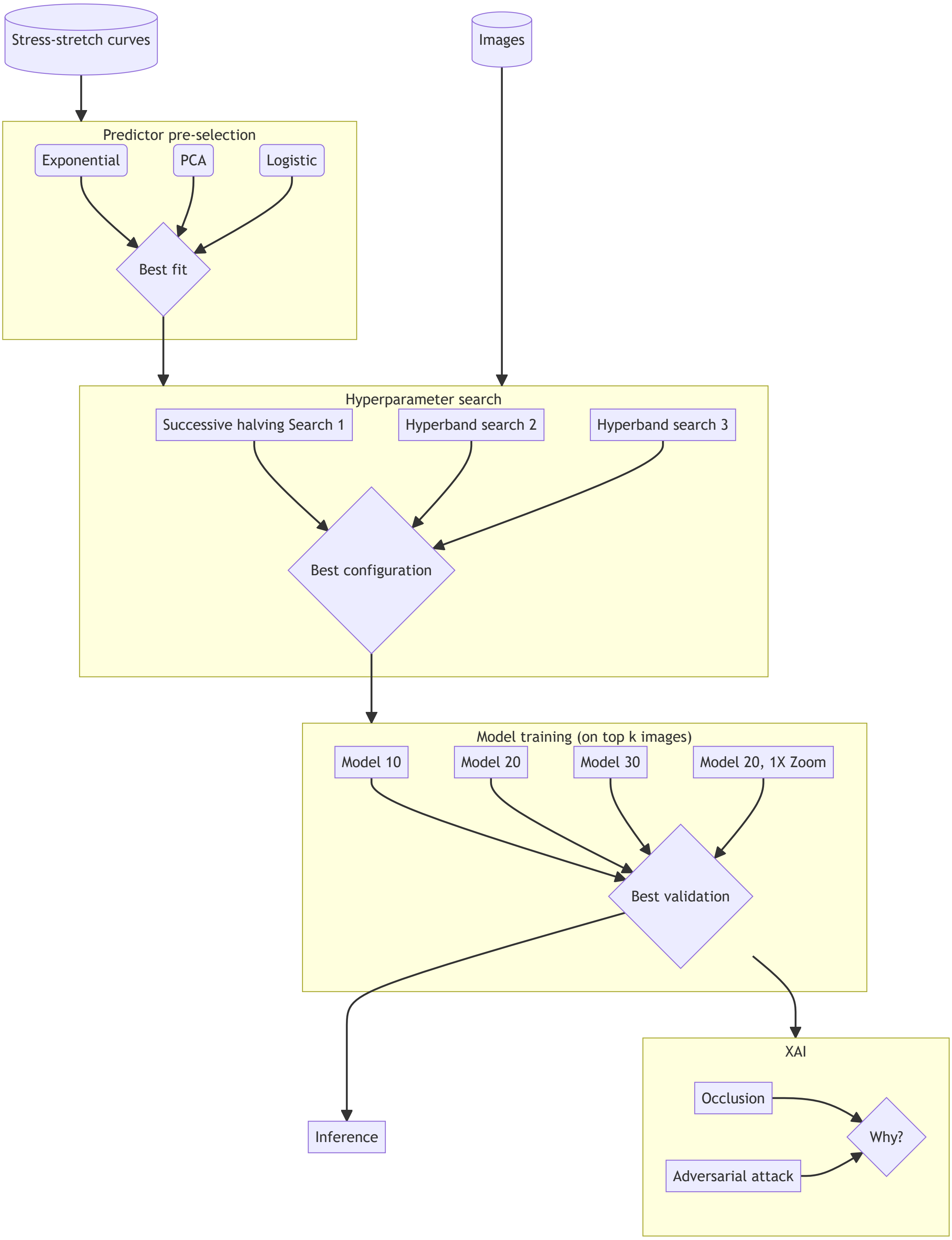


Figure 3.3: Skinstression development flow. Exponentials, principal component analysis (PCA) reconstructions, and logistic curves were fit to the stress-strain curves. The best-fitted model was used to create training targets. The targets and images were used as input for further training. The hyperparameter search was performed by four Hyperband studies and the best configuration was used to train four models. The best validation loss determines the final model. The final model is tested and interrogated using occlusion and an adversarial attack.

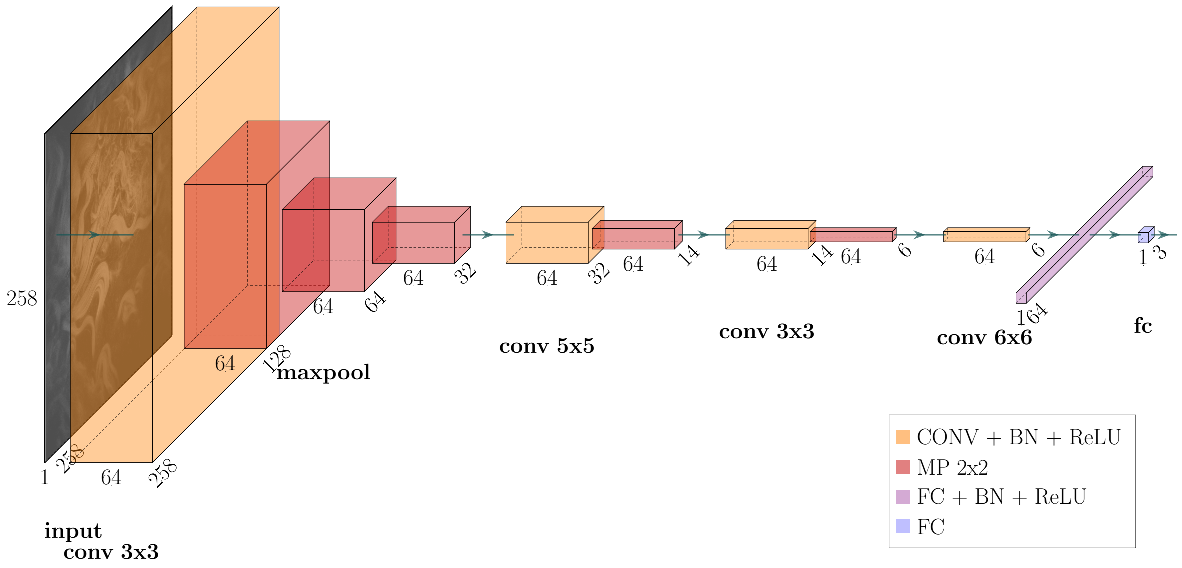


Figure 3.4: The convolutional neural network consists of five blocks. The first four blocks contain convolution, maxpooling, and batch normalization layers. The last block contains a fully connected network. It requires an input of 258 × 258 to get a vector of length 3 as output.

by eye. A fit is considered good if it passes reasonably through all data points and has few inflection points.

Only if PCA on the full dataset works reasonably well, it is possible to use PCA on a subset and use it to reconstruct another subset. This would be useful if PCA was used to construct predictors, as using PCA results of the full dataset introduce information leakage from the test sets to the training set, because the components describe data from both subsets. This is unlike Ref. [31] where information leakage was not considered.

[31]: Soylu (2022), *Developing a non-invasive approach to estimate physical parameters of skin tissue from HHG microscopy using convolutional neural networks*

Convolutional neural network

The basis of the model originates from Liang *et al.* [42] and is adapted by Soylu [31]. The model, a convolutional neural network, consists of five blocks. The first block consists of a convolutional layer with a 3×3 kernel, taking in one channel and have 64 channels as output. The output is then normalized per batch using BN (Subsection 2.4.1). The normalized batch is passed through a ReLU (Subsection 2.3.4) layer. After activation, three 2×2 maxpool (Subsection 2.3.3) layers are applied. The next second block is like the first block, but with a 5×5 convolution kernel and just one maxpool layer. The third block is like the second block, but with a 3×3 convolution kernel. The fourth block is like the second and third block, but with a 6×6 and without a maxpool layer.

[42]: Liang et al. (2017), *A deep learning approach to estimate chemically-treated collagenous tissue nonlinear anisotropic stress-strain responses from microscopy images*

[31]: Soylu (2022), *Developing a non-invasive approach to estimate physical parameters of skin tissue from HHG microscopy using convolutional neural networks*

The fifth block flattens the input and consists of a two linear layers. The first linear layer maps 64 nodes to N_{nodes} nodes. After the first linear layer, BN and ReLU activation is applied. The second linear layer maps N_{nodes} nodes to 3 nodes. A linear activation function ensures the output is continuous and unaltered. The model is shown in Figure 3.4.

The dropout layers in [31] are replaced by BN layers, as the input is not

[31]: Soylu (2022), *Developing a non-invasive approach to estimate physical parameters of skin tissue from HHG microscopy using convolutional neural networks*

normalized and studies report better performance with BN. Bias of all layers preceding BN layers have been set to zero to remove redundancy.

The neural network weights are initialized according to the method described by He et al. [43], using a uniform transform.

[43]: He et al. (2015), *Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification*

Hyperparameter optimization

First, benchmark search 1 was done using Successive Halving with 100 trials. See Appendix A.3 for a summary of configuration search space $\mathscr{C}$. To allow trials to warm up, a minimum of 100 epochs were allowed. To limit the trial duration, a maximum of 3000 epochs were allowed. The number of trials were reduced with a reduction factor of $\eta = 4$. Trial parameters were sampled using the non-multivariate TPE algorithm.

The optimization was performed with the Adam optimizer ($\beta_1 = 0.9$, $\beta_2 = 0.999$), weighted focal MSE loss, and a cosine annealing with warm restarts learning rate scheduler. Every trial used the complete dataset after data preparation (Subsection 3.3.1).

The search uses a few data augmentations that are assumed to not alter the physical context of the image. That is, force is exerted on the tissue unilaterally, which is horizontal in the image. Therefore, flipping the image either vertically or horizontally is assumed to not change the stretch behavior. Both flipping operations occur with a probability of 0.5. Moreover, the images' intensity is randomly changed uniformly by 0.7 % to 1.3 %.

To possibly find a more optimal set of hyperparameters, two searches with the Hyperband algorithm with 300 trials was performed. Trial parameters were sampled using the multivariate TPE algorithm. The learning rate was warmed up linearly for the first 20 epochs.

Search 2 introduces random 700 × 700 cropping to further artificially increase the number of available images to train on. Search 3 includes the Yeo-Johnson transformation to see how input normalization affects the training outcome.

For a summary of the hyperparameter searches, see Table 3.1.

Algorithms provided by Optuna [44] were used to choose trial configurations and keep track of trials.

[44]: Akiba et al. (2019), *Optuna: A Next-Generation Hyperparameter Optimization Framework*

Training

The AI was trained on a with LDS smoothed target variable distribution. The targets were weighted with the inverse square root, to limit the impact of LDS. Using the best configuration from hyperparameter optimization, a model is trained further for a total of 10000 epochs. The learning rate scheduler was cosine annealing with warm restarts to allow for model ensembling later during inference[45], if deemed useful. The learning rate was warmed up linearly for the first 20 epochs. To see the influence of image quality, the model is trained using an ordered set of images. The images are ordered with maximum entropy first and the model is trained on all images and the top 10, 20 and 30 images of every stack. Moreover, to see the effect of using the original zoom level with the greatest detail at

[45]: Huang et al. (2017), *Snapshot Ensembles: Train 1, Get M for Free*

Search	1	2	3
CLAHE	✓	✓	✓
Yeo-Johnson transform	✗	✗	✓
Random 700 × 700 cropping	✗	✓	✓
Intensity jitter	✓	✓	✓
Random horizontal flip	✓	✓	✓
Random vertical flip	✓	✓	✓
Random sharpness	✗	✗	✗
Random gaussian blur	✗	✗	✗
Random rotation	✗	✗	✗
Search algorithm	SH	HB	HB
Multivariate TPE	✗	✓	✓
trials	100	300	300
learning rate warmup	✗	✓	✓

Table 3.1: Summary of settings for hyperparameter searches performed. Hyperparameters are grouped by operation type (image preprocessing, image augmenting, target weighting) and in applied order. Every search is done with the search space described in Table A.1.

hand, the best 20 images of every stack were center-cropped to 500 × 500 and further randomly cropped to 258 × 258. The lowest validation focal loss is used to compare model performance. The model with the lowest validation loss is used for testing.

Pytorch [46] was used to perform automatic differentiation on NVIDIA GeForce RTX 2070 Super GPUs on the BAZIS high performance computing cluster.

[46]: Paszke et al. (2019), *PyTorch: An Imperative Style, High-Performance Deep Learning Library*

Internal validation

The thigh dataset is randomly distributed into a training (64 %), validation (16 %), and test (20 %). The distribution is stratified by person, meaning samples corresponding to the same person cannot live in two subsets simultaneously. The AI learns from the training set. Every iteration, it is validated against the validation set. During inference, the AI is applied to the test set as external validation. The prediction is assessed by comparing it with measured strain-stress curves where R^2 is calculated with a 95 % confidence interval.

Bias study

It is important to perform a study on bias for possible explanations of varying AI performance. The samples were taken from individuals with varying age and gender. Moreover, from some individuals, more mechanical measurements were taken. Therefore, the age, gender, and number of samples were summarized.

3.3.3 Explainability

Occlusion

To explain the model output, occlusion Subsection 2.6.1 is used. Using the Captum Python library [47], occlusion is done with 3 × 3 square patches moving with strides of 1. All values in the patch have been replaced with 0.

[47]: Kokhlikyan et al. (2020), *Captum: A unified and generic model interpretability library for PyTorch*

Adversarial attack

To see the importance of homogeneous tissue within the skin tissue, black patches have been filled using Fiji [48]. Filling is done by drawing full white ellipses on top of the original image. Moreover, in another attack, a black patch is copied to other locations in the image.

[48]: Schindelin et al. (2012), *Fiji: An open-source platform for biological-image analysis*

3.4 Results

3.4.1 Participants

Earlier studies [29, 31, 35] include 18 individuals (5 men, 4 women, and 6 unknown). Abdomen data was excluded, because the strain-stress curves differ significantly from the thigh. All thigh data is included, which is different from the original study, where only the 48 latest samples were used. These considerations result in data including 15 individuals (5 men, 4 women, 3 unknown). Ages range from 61 to 94. From every skin tissue piece, measured strain-stress curves are shown in Figure 3.5. The number of measured strain-stress curves range from 1 to 13. The source of data is summarized in Figure 3.6.

[29]: Zhou et al. (n.d.), *Three-dimensional Characterization of Mechanical Properties and Microstructures of Human Dermal Skin*
[31]: Soylu (2022), *Developing a non-invasive approach to estimate physical parameters of skin tissue from HHG microscopy using convolutional neural networks*
[35]: Haasterecht et al. (2023), *Visualizing dynamic Three-dimensional changes of human reticular dermal collagen under mechanical strain*

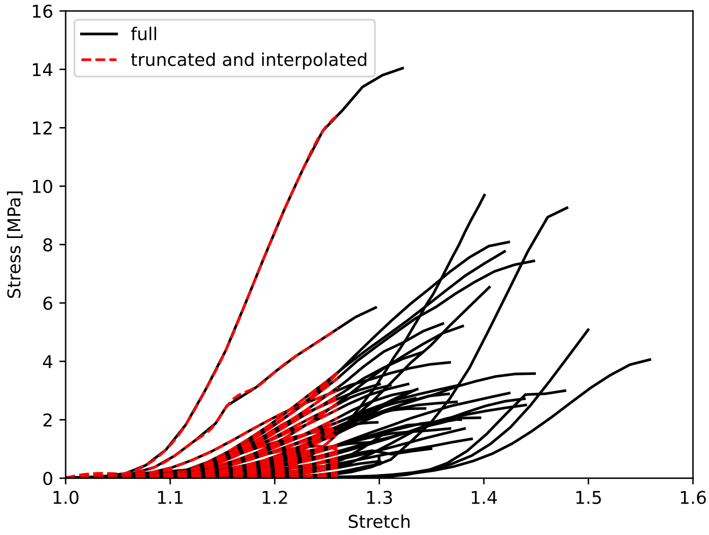


Figure 3.5: The strain-stress curves (black) were truncated and interpolated (dotted red) using non-uniform, interpolating splines on the stretch values of the curve with the lowest maximum stretch.

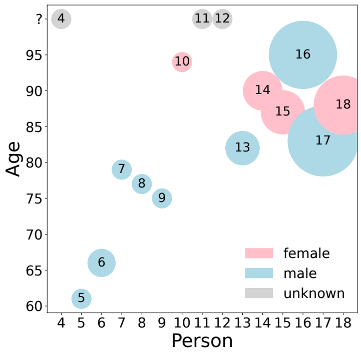


Figure 3.6: The selected individuals and their sex, age and number of strain-stress curves. Bubble size and centered text show number of curves.

3.4.2 Predictor pre-selection

Exponential

An exponential fit to Equation 3.5 is shown in Figure 3.7. $R^2 = 0.9497$. The exponent is not able to fit the plateau that is often exhibited near maximum stretch.

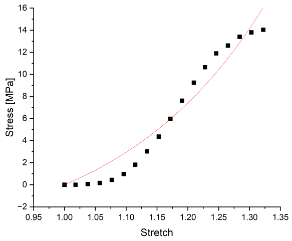


Figure 3.7: Exponential fit with Equation 3.5 (red) for one stress-strain curve (black). Fit parameters were $G_0 = (23.8 \pm 4.0)$ MPa unit stretch^{-1}, and $\lambda = (4.1 \pm 1.0)$ unit stretch. $R^2 = 0.9497$.

PCA

The PCA fit for every truncated and interpolated strain-stress curve is depicted in Figure A.1. An example is shown in Figure 3.8. For every fit, R^2 is calculated with respect to the interpolated and truncated data. On average, $\overline{R^2} \approx 0.9926$ (SE 0.0003). Due to the nature of PCA, the exponential part of the curves that rise later is not included in the making of the fit.

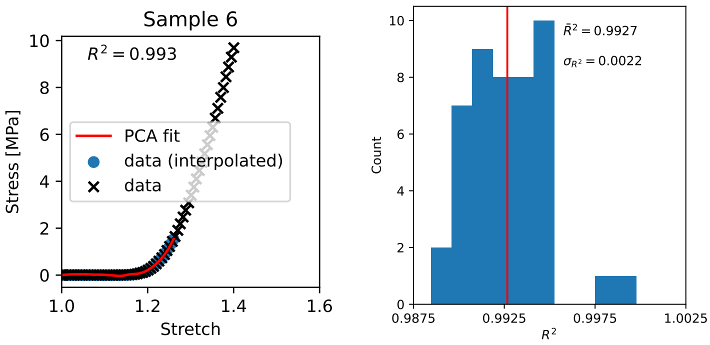


Figure 3.8: Top panel: PCA fit (red) to one stress-strain curve (black). $R^2 = 0.993$. Bottom panel: distribution of R^2 on all stress-strain curves. Mean $R^2 = 0.9927$ (red) with a standard deviation of 0.0022.

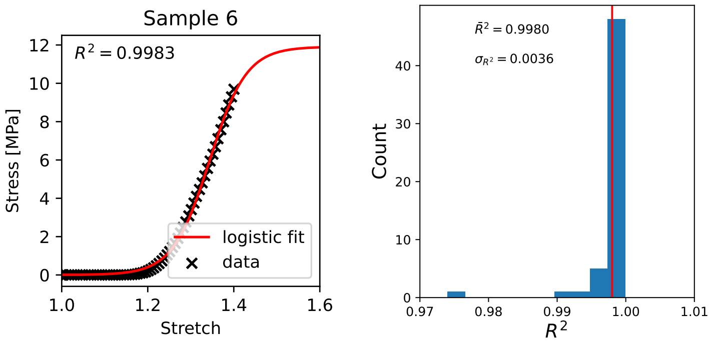


Figure 3.9: Top panel: Logistic curve fit (red) to one stress-strain curve (black). $R^2 = 0.9983$. Bottom panel: distribution of R^2 on all stress-strain curves. Mean $R^2 = 0.9980$ (red) with a standard deviation of 0.0036.

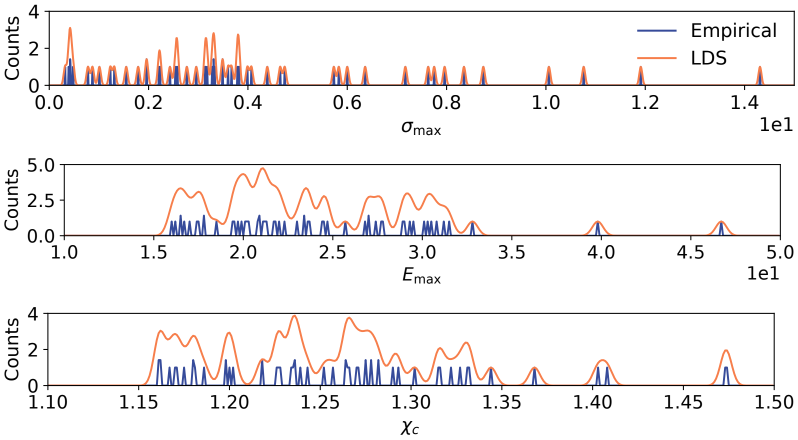


Figure 3.10: Empirical target distribution (red) and label density smoothed (LDS) target distribution (blue) for $\sigma_{\max}$, $E_{\max}$, and χ_c. LDS aims to fill the gaps between the actual measurements. LDS also extrapolates target distributions. For all targets, the kernel size was 30 units, and the standard deviation 3 units. The bin width were 0.01, 0.1, and 0.001 units for $\sigma_{\max}$, $E_{\max}$, and χ_c, respectively.

Logistic curve

The logistic curve fit for every strain-stress curve is shown in Figure A.2. An example is shown in Figure 3.9. For every fit, R^2 is calculated. On average, $\overline{R^2} \approx 0.9984$ (SE 0.0002). Because the logistic curve describes the stress-strain data more accurate than PCA or the exponential, it will be used from now on.

Label density smoothing

The labels were smoothed using LDS using Gaussian kernels. All kernels had a size of 30 and a standard deviation of 3. The distribution bins were manually set to $(\sigma_{\max,\min}, \sigma_{\max,\max}, s_{\sigma_{\max}}) = (0, 15, 0.01)$MPa, $(E_{\max,\min}, E_{\max,\max}, s_{E_{\max}}) = (0, 50, 0.1)$MPa unit stretch^{-1}, and $(\gamma_{c,\min}, \gamma_{\max,\max}, s_{\gamma_{\max}}) = (1, 1.5, 0.001)$unit stretch, where s is the bin width. The smoothed distribution is shown in Figure 3.10.

3.4.3 Image quality

Presumably, the model will yield erroneous predictions if images of bad quality are provided. Moreover, the measurement was not always done in the same order, *i.e.* from the lowest to the highest tissue or vice versa. Therefore, two image quality measurements were explored: Shannon entropy and kurtosis.

The kurtosis of some image stacks were used to validate the image quality by eye, see Figure 3.11. The entropy of the same image stacks as for kurtosis were used to validate the image quality by eye, see Figure 3.12. For reasons explained in Subsection 3.5.2, entropy is further used to choose the top quality images from every stack.

3.4.4 Hyperparameter optimization

The loss curves of the best trials of every search are shown in Figure 3.13 Both training and validation loss show an increase at the learning rate restarts.

3.4.5 Training

Training was done on the best performing set of hyperparameters, which was found by search 1. Search 1 yielded the lowest validation loss. The corresponding hyperparameters in Table 3.2 were used for further training.

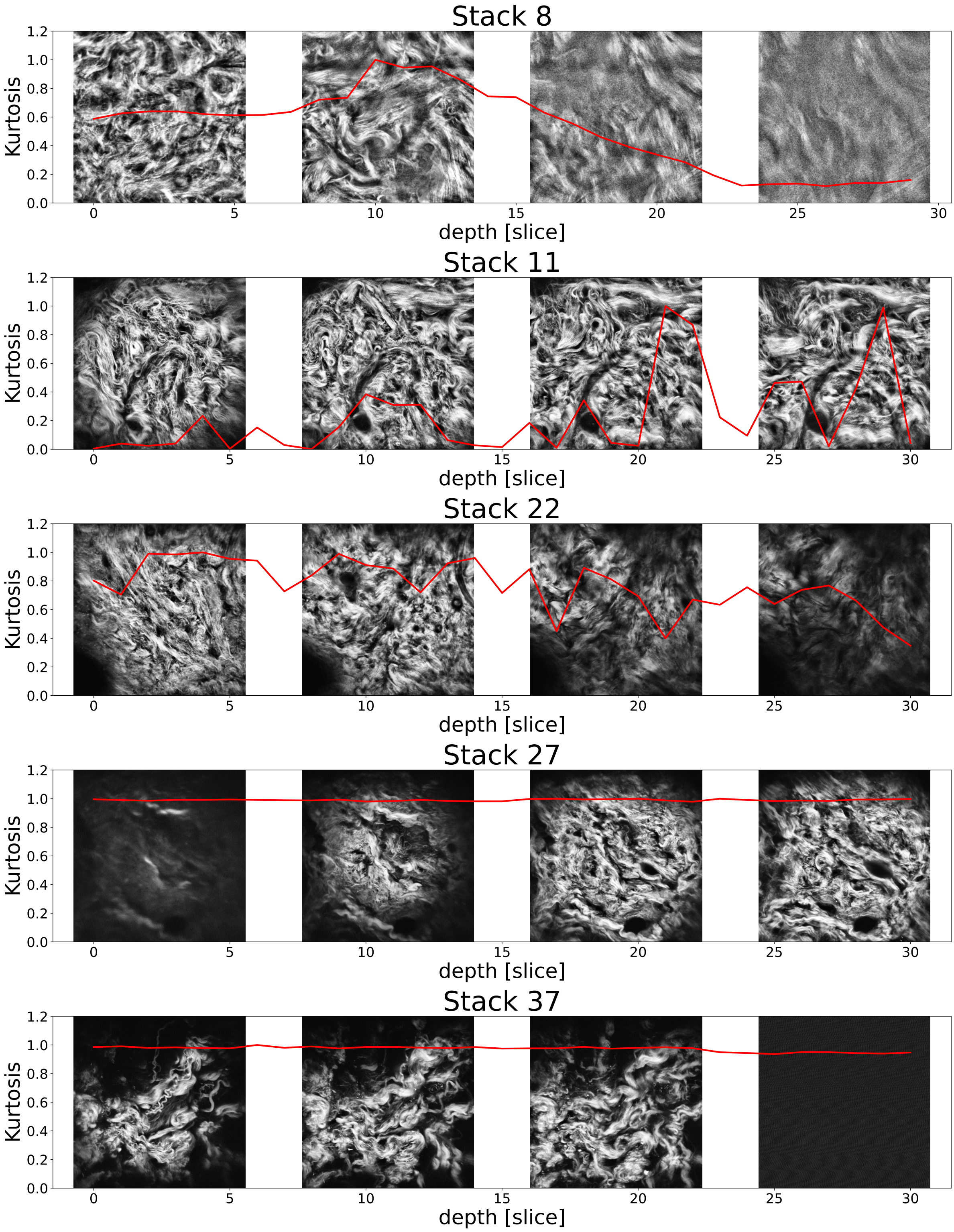


Figure 3.11: Kurtosis as a function of image stack depth for a representing subset. Four images which are equally spaced in physical units are inset.

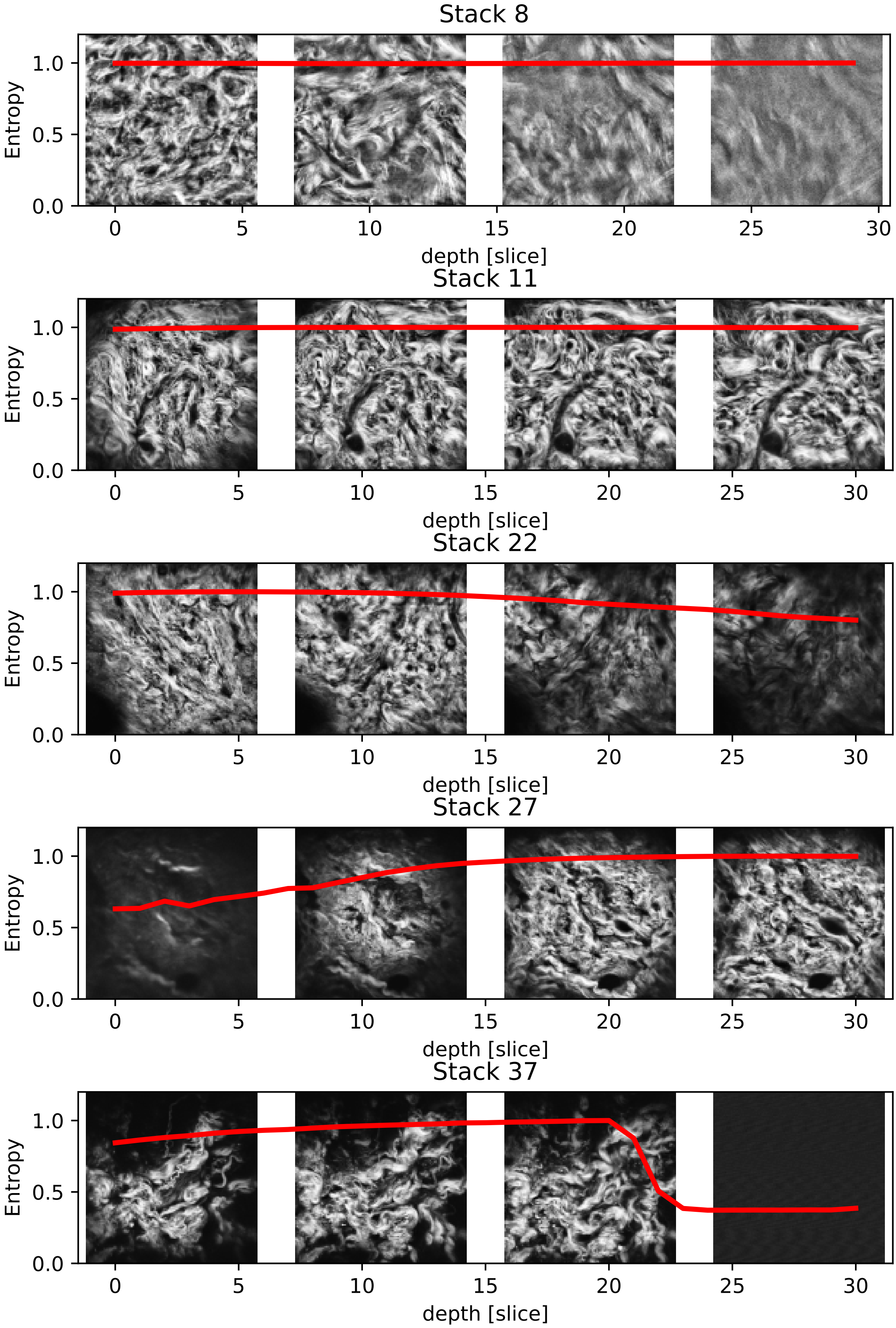


Figure 3.12: Normalized Shannon entropy as a function of image stack depth for a representing subset. Four images which are equally spaced in physical units are inset.

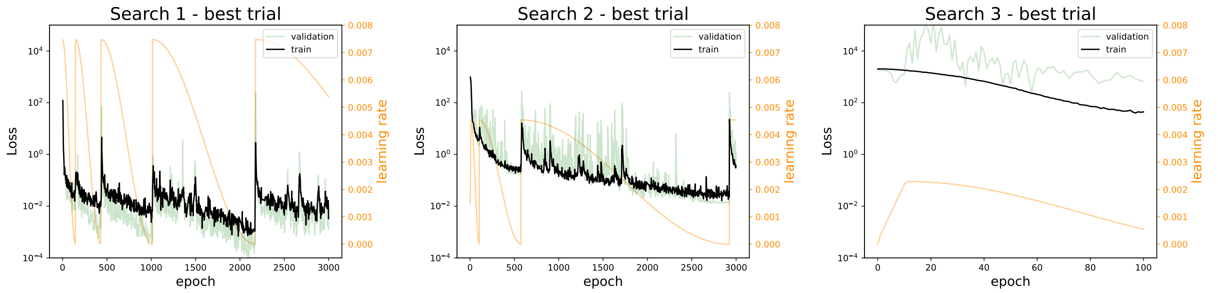


Figure 3.13: Successive halving trial 71 of 100 has shown the best loss for search 1. Hyperband trial 184 of 300 has shown the best loss for search 2. Hyperband trial 73 of 300 has shown the best loss for search 3. The training loss (black) and validation loss (light green) are shown. The learning rate (orange) restarts explain sudden increase in loss.

Three trainings were performed. For every training, the 10, 20, and 30 images of every stack with the highest entropy were selected to exclude any noisy images. Also, using the 20 best images of every stack, the model is trained without scaling of the original image. The loss curves are shown in Figure 3.14. The lowest validation losses are shown in Table 3.3. Overall, the model trained with 20 images shows the lowest total loss. This model will be used during testing.

To verify performance of training, the stress-strain curves of a random training batch are plotted with the raw data in Figure A.3. For this batch, $\overline{R^2} = 0.75$ (SE 0.13).

Table 3.2: Skinstression configuration used during training. Parameters are ordered by their importance, calculated with fANOVA. LR, WD, and BS are learning rate, weight decay and batch size, respectively.

Param.	Value	Imp.
LR	0.00747	0.41
T_0	145	0.24
WD	2.97×10^{-5}	0.22
n_{nodes}	64	0.08
T_{mult}	2	0.05
BS	16	0.00

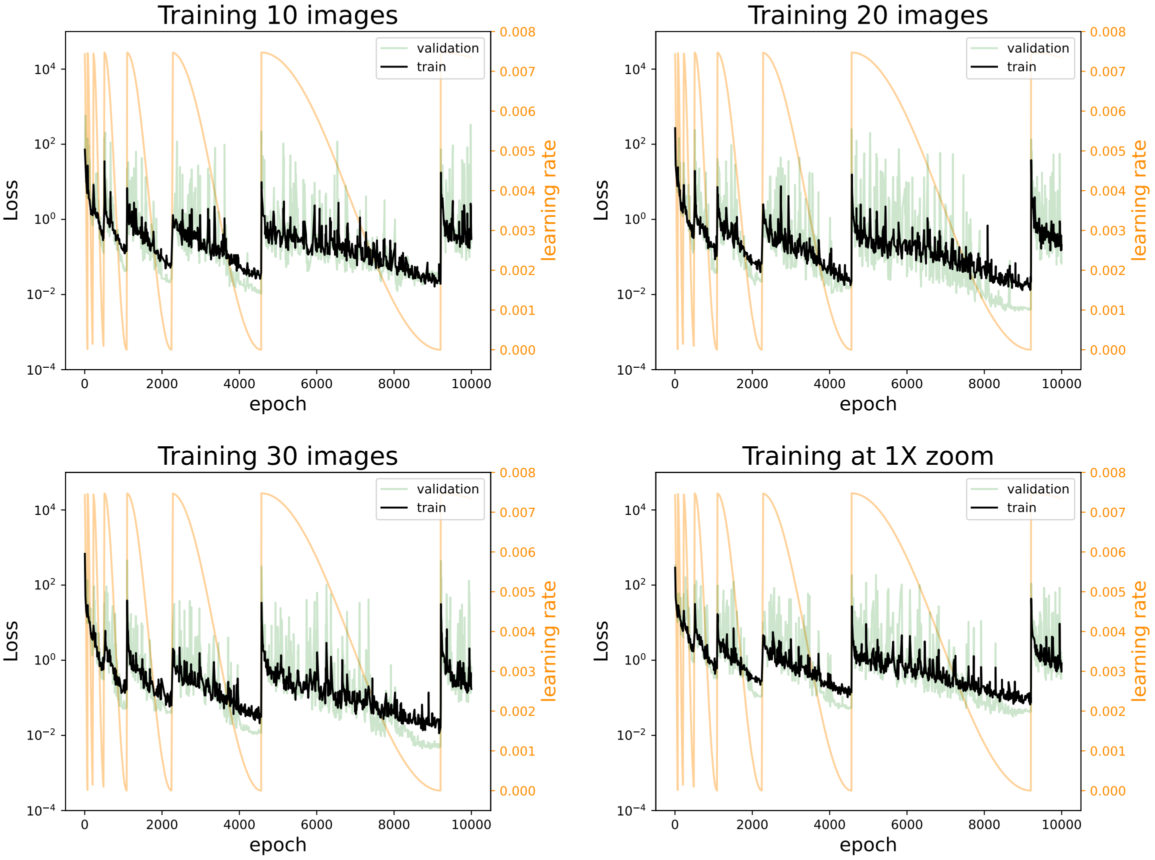


Figure 3.14: Training losses when using 10, 20, and 30 images as well as a training using 20 images, but with unscaled images. The training loss (black) is followed by the validation loss (light green). The learning rate (orange) restarts explain sudden increase in loss.

Table 3.3: The lowest validation loss per N_{best} images. The training using 20 images has the lowest validation loss. * denotes training without rescaling.

# Images	Validation loss
10	0.0097
20	**0.0035**
20*	0.27
30	0.0040

3.4.6 Testing

The results of the best performing images in the test set are shown in Figure 3.15. The higher R^2, the better the performance. Over the best performing images of every sample, a low $\overline{R^2} = -0.36$ (SE 0.60).

The performance difference between the training and test set is significant. The donors of the test set are both male and female with ages comparable to the training set (see Subsection 3.4.1). However, most samples of the test set originate from the same person. If the skin of this donor exhibits other behavior, it might be difficult for the AI to predict well.

3.4.7 Model specification

The foundational neural network is described in Figure 3.4. The model with the lowest validation loss was chosen as final model. It accepts a batch of images as input and outputs a vector $(i, \sigma_{\max}, E_{\max}, \gamma_c)$, where i denotes the ith image of the batch.

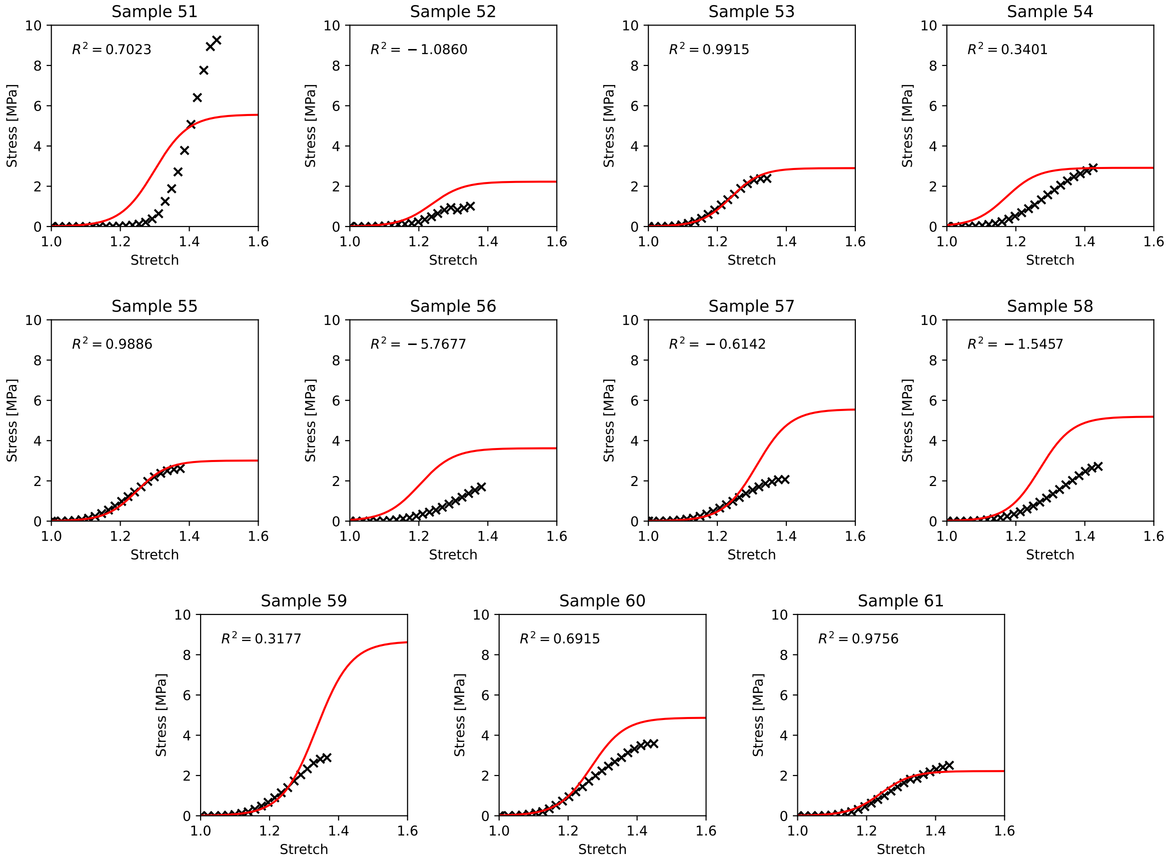


Figure 3.15: Results of the best performing image per sample are shown. Performance is quantified by R^2.

3.4.8 Usability

The prediction AI may be used for two purposes. One purpose is to skip the mechanical measurement on skin tissue and immediately get an estimation of stretch properties. Another is to match stretch information to physical features present in the collagen fiber networks. The model should not be used as validation for mechanical measurements, as it is trained and tested on a small dataset.

Input data should be assessed on quality. A measure of quality is Shannon entropy, which is used in this study. However, the model may benefit from other quality measures. The user should experiment what measure works best. The application does allow to only select the top N_{best} images.

The model only accepts batches of two-dimensional 500 × 500 px images. Currently, the user should separate any higher-dimensional data and crop 0.2 mpp images to fit the model.

3.4.9 Explainability

Occlusion

Algorithmically occluded images have been presented to the model to obtain attributions at the pixel level. For a summary of the test set, they are shown in blended heatmaps in Figure 3.17. Figure 3.18 shows attribution maps for all test images of sample 53.

Attributions appear to coincide with collagen fibers. This is important, as the abundance of collagen is hypothesized to relate to the stretch properties. It is unclear how positive and negative attributions give insight into the prediction of stretch properties. It is important to note that attribution maps belonging to the best performing predictions give the most appropriate insight into the AI reasoning.

The heatmaps give limited insight into the performance errors between samples across the test set or images within a sample. There is no clear relation between the attribution mean and/or standard deviation and R^2.

Adversarial attack

Figure 3.19 shows attribution maps for all outputs to compare attacked images with the original image. Figure 3.16 shows the predicted logistic curves for the attacked images and the original image. These results show that artificially filling holes increases the stiffness and resistance, while adding holes decreases stiffness and resistance.

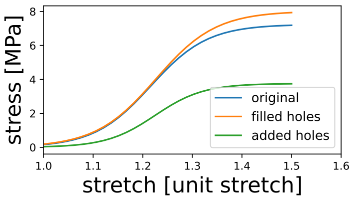


Figure 3.16: Prediction output after hole attack.

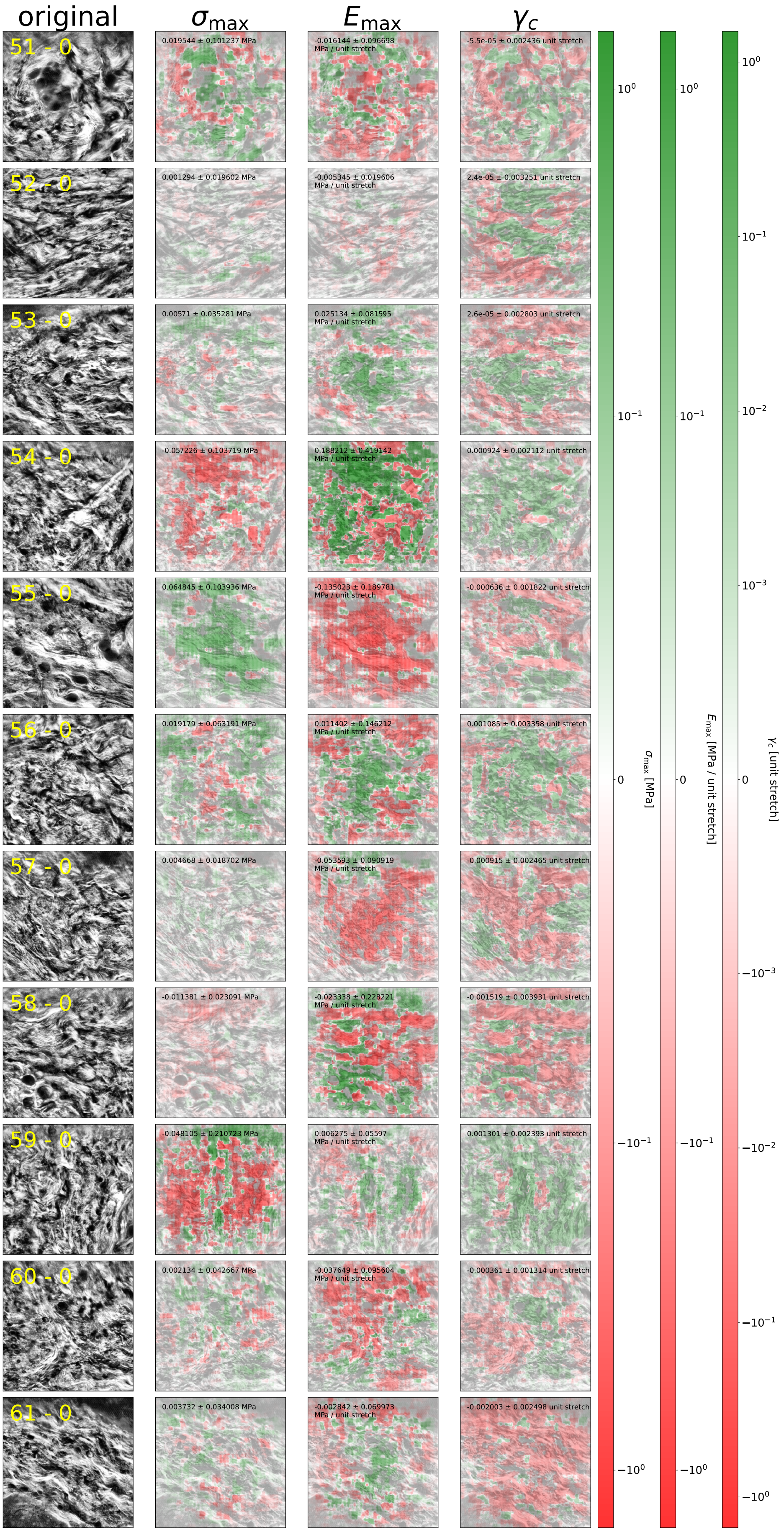
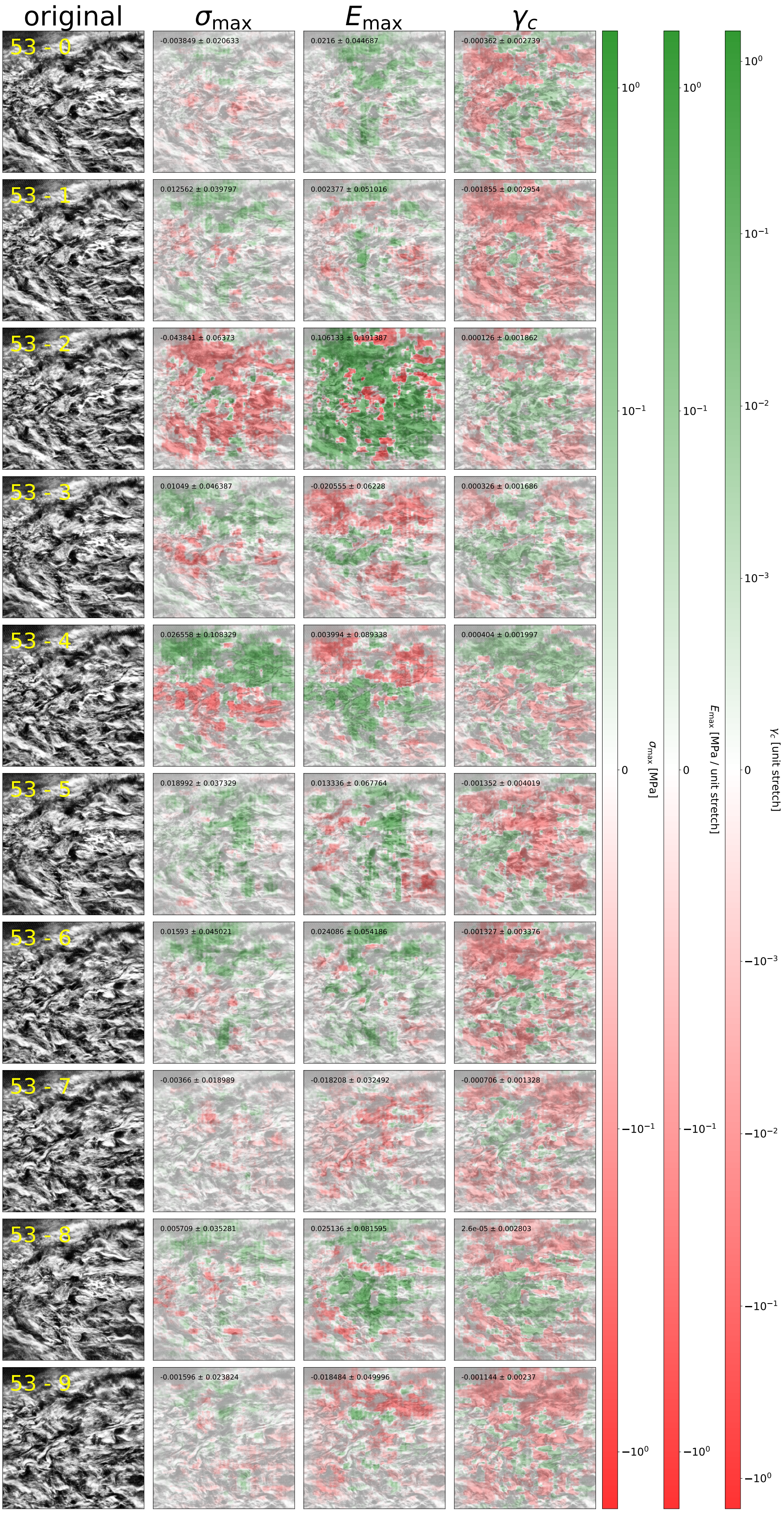

original
$\sigma_{\max}$
$E_{\max}$
γ_c
51 - 0
0.019544 ± 0.101237 MPa
-0.016144 ± 0.096698 MPa / unit stretch
-5.5e-05 ± 0.002436 unit stretch
52 - 0
0.001294 ± 0.019602 MPa
-0.005345 ± 0.019606 MPa / unit stretch
2.4e-05 ± 0.003251 unit stretch
53 - 0
0.00571 ± 0.035281 MPa
0.025134 ± 0.081595 MPa / unit stretch
2.6e-05 ± 0.002803 unit stretch
54 - 0
-0.057226 ± 0.103719 MPa
0.188212 ± 0.419142 MPa / unit stretch
0.000924 ± 0.002112 unit stretch
55 - 0
0.064845 ± 0.103936 MPa
-0.135023 ± 0.189781 MPa / unit stretch
-0.000636 ± 0.001822 unit stretch
56 - 0
0.019179 ± 0.063191 MPa
0.011402 ± 0.146212 MPa / unit stretch
0.001085 ± 0.003358 unit stretch
57 - 0
0.004668 ± 0.018702 MPa
-0.053593 ± 0.090919 MPa / unit stretch
-0.000915 ± 0.002465 unit stretch
58 - 0
-0.011381 ± 0.023091 MPa
-0.023338 ± 0.228221 MPa / unit stretch
-0.001519 ± 0.003931 unit stretch
59 - 0
-0.048105 ± 0.210723 MPa
0.006275 ± 0.05597 MPa / unit stretch
0.001301 ± 0.002393 unit stretch
60 - 0
0.002134 ± 0.042667 MPa
-0.037649 ± 0.095604 MPa / unit stretch
-0.000361 ± 0.001314 unit stretch
61 - 0
0.003732 ± 0.034008 MPa
-0.002842 ± 0.069973 MPa / unit stretch
-0.002003 ± 0.002498 unit stretch
10^0
10^{-1}
0
-10^{-1}
-10^0
$\sigma_{\max}$ [MPa]
10^0
10^{-1}
0
-10^{-1}
-10^0
$E_{\max}$ [MPa / unit stretch]
10^0
10^{-1}
10^{-2}
10^{-3}
0
-10^{-3}
-10^{-2}
-10^{-1}
-10^0
γ_c [unit stretch]

Figure 3.17: Original image and attribution heat maps. Attributions are blended with the original image. Attributions were calculated using 20 × 20px occlusion. Color scale is from -3 to 3 units, and is linear from -0.1 MPa to 0.1 MPa, -0.1 MPa unit stretch^{-1} to 0.1 MPa unit stretch^{-1}, and -0.001 unit stretch to 0.001 unit stretch. Mean ± standard deviation are inset. Zoom in the digital version for details.



Continued on next page.

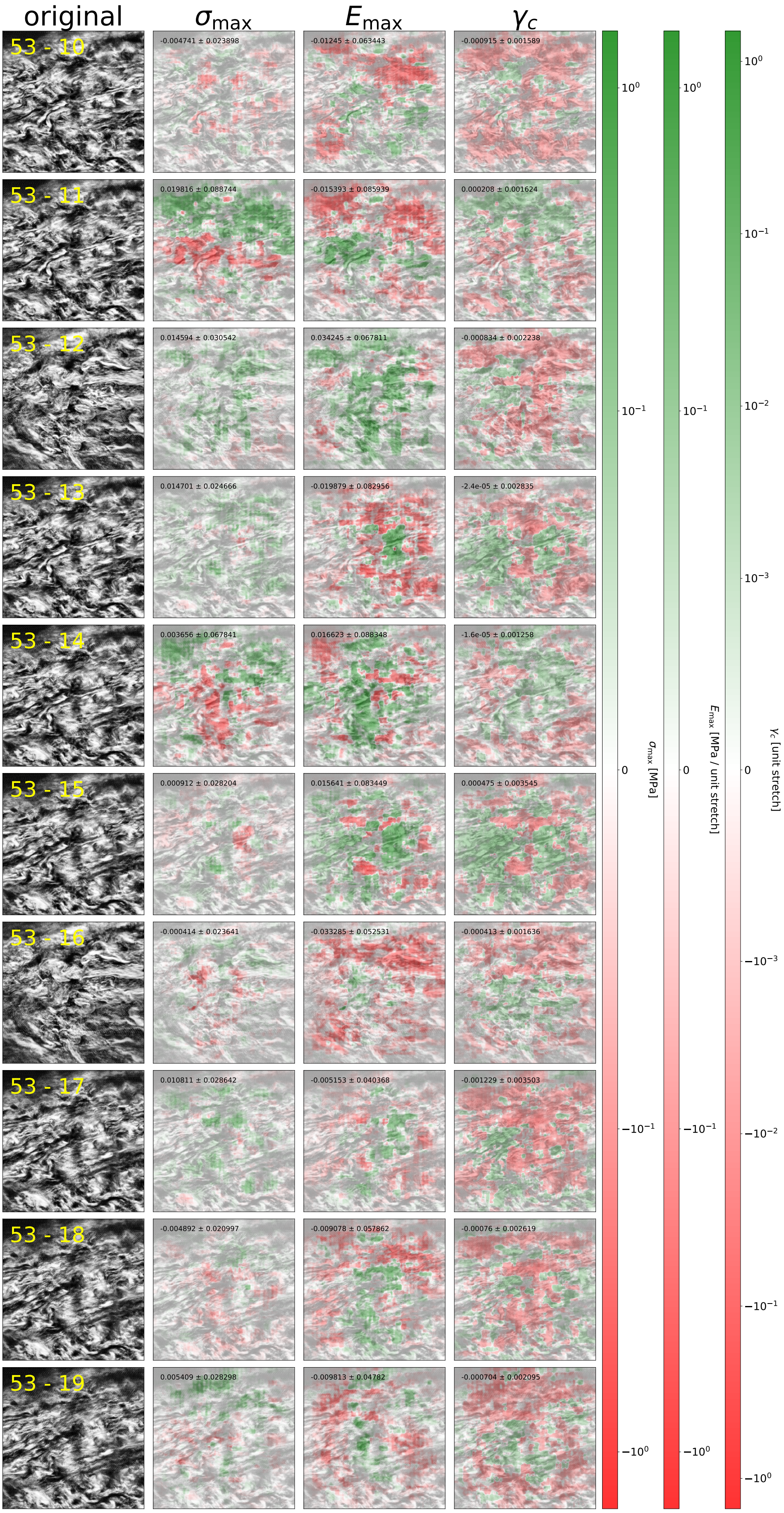


Figure 3.18: Original image and attribution heat maps for sample 53. Attributions are blended with the original image. Attributions were calculated using 20×20px occlusion. Green and red show which parts attribute to under- and overestimations, compared to output calculated from an unoccluded image. Color scale is from -1.5 to 1.5 units, and is linear from -0.1 MPa to 0.1 MPa, -0.1 MPa unit stretch^{-1} to 0.1 MPa unit stretch^{-1}, and -0.001 unit stretch to 0.001 unit stretch. Mean ± standard deviation are inset. Zoom in the digital version for details.

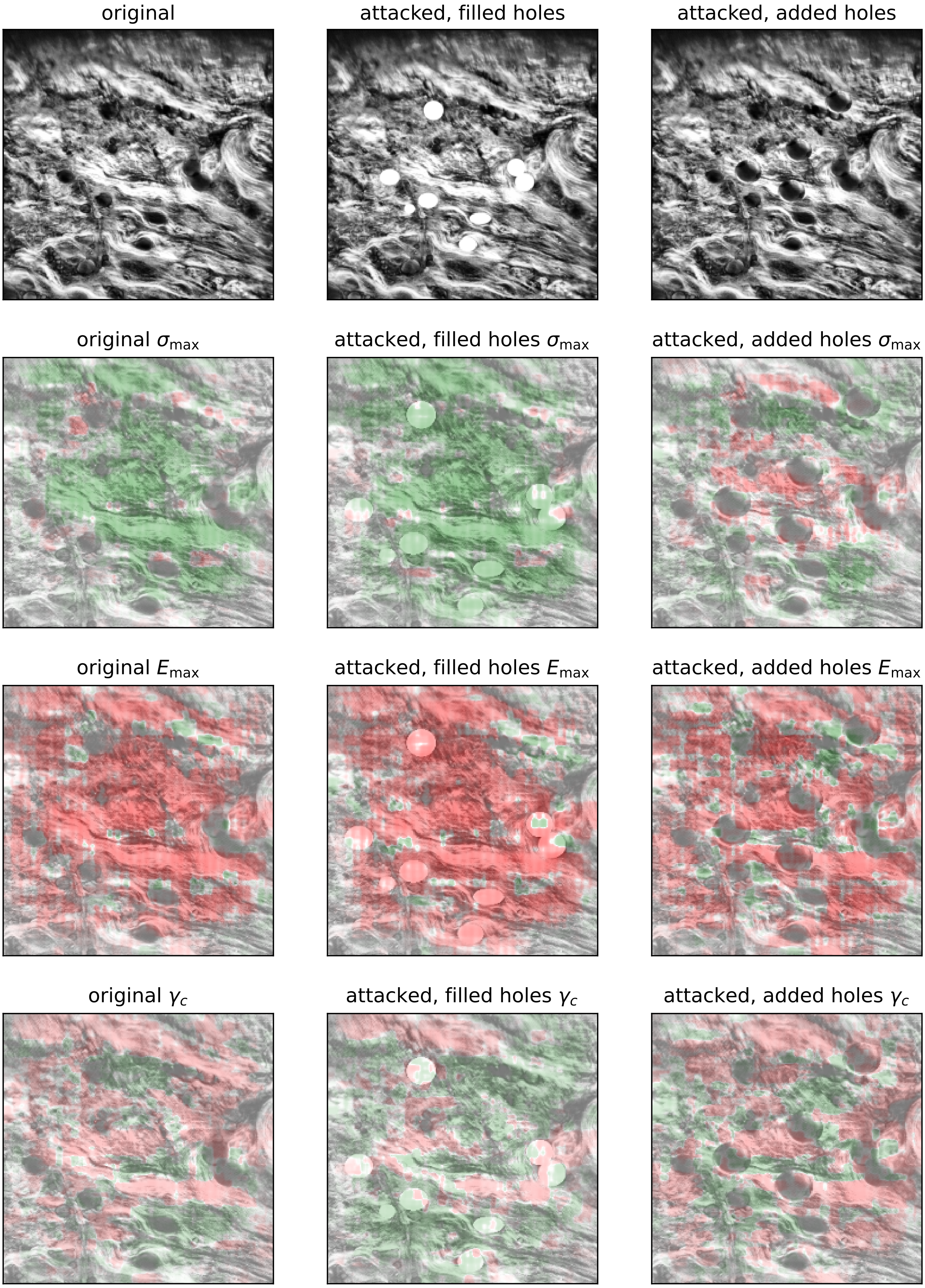

original
attacked, filled holes
attacked, added holes
original σ_{max}
attacked, filled holes σ_{max}
attacked, added holes σ_{max}
original E_{max}
attacked, filled holes E_{max}
attacked, added holes E_{max}
original γ_c
attacked, filled holes γ_c
attacked, added holes γ_c

Figure 3.19: Top row shows a visualization of the two attacks. The middle image has holes filled with white. The right image has duplicates of one hole. The next rows show occlusion attribution maps for all outputs. Color scales are equal to Figure 3.18.

3.5 Discussion

3.5.1 Logistic curve fits stress-strain curves better than an exponential or PCA

With an $\overline{R^2} \approx 0.9984$ (SE 0.0002), the logistic curve fits stress-strain data (Figure A.2) better ($p < 0.01$) than the exponential or PCA ($\overline{R^2} \approx 0.9926$ (SE 0.0003), Figure A.1). The exponential does not fit the plateau that is often present at the end of a stress-strain curve (Figure 3.7). This plateau is essential to the skin tissue dynamics. It shows at which point the integrity of the collagen matrix breaks down, *i.e.* when the skin cannot resist the force acting on it. PCA had to be done after truncating all curves to maximum strain of the curve with the smallest maximum strain. This resulted in PCA fits that only described the first region of the curve. Moreover, to fairly create PCA fits, PCA has to be trained on a training set and values extracted from the test set must be projected onto this test set. This may lead to generalizability problems. For these reasons, the logistic curve parameters are used as a predictors.

3.5.2 Shannon entropy is a better measure to exclude noise in collagen images than kurtosis

Kurtosis and Shannon entropy were used to exclude noise in SHG image of collagen. For rather homogeneous images like stack 8, kurtosis is able to characterize the fogginess, while entropy finds images of equal quality. However, for well-structured tissue like stack 11, 12, 13 and 4, entropy recognizes faint images and qualifies them as bad. This is particularly useful for images that did not include any useful information (like stack 37). Kurtosis does not succeed to distinguish dark images from bright ones. Kurtosis also has the tendency to fluctuate, which is unexpected between subsequent slices. For these reasons, Shannon entropy has been used to choose the top N_{best} images.

3.5.3 The model does not generalize well

Although the train results were promising, the test results of Figure 3.15 show a lack of generalizability. Only two images from different stacks of the test set yield a stress-strain curve with $R^2 = 0.99$. This shows that the training set does not include enough features similar to the features in the test set. As most of the images in the test set belong to one person, it may be possible that this person has exceptional skin tissue compared to samples in the training set. More images need to be included in the training set for the model to generalize well to held-out test cases.

3.5.4 Artificially increasing collagen density increases maximum stress prediction and stiffness and vice versa

As shown in Figure 3.16, artificially filling holes with collagen increases the maximum stress significantly and increases the maximum Young's modulus slightly. This is expected, as an increase in collagen density generally is an increase in the number of springs in the system. Adding springs increases the stiffness. The increase in maximum stress is likely due to the addition of springs supporting each other dividing the load. On the contrary, with comparable reasoning, artificially decreasing the collagen density by duplicating a hole lowers both the maximum stress and Young's modulus.

3.5.5 Future studies

Perform cross validation and increase data variance

In this study, no cross validation is performed. Training a network on different training subsets might increase the performance on the test set. The test set might include patterns that were not present in the training set and therefore will not activate critical parts of the neural network. The split between training and test sets should be made such that the training set has high variance and is thus a reasonably good estimate of the whole population, without knowing test set images.

Therefore, the training set should also include images from human skin that is damaged in any way. For example, damaged tissue is caused by smoking [49] or wounds that left behind scars with an increased tensile strength [50]. Moreover, aging drastically impacts skin tissue integrity. It is unknown if stretch of young skin tissue is predicted well by the neural network. Therefore, the neural network should only be used to predict from old skin tissue.

Skin tissue from other body parts might show different stretch properties. Therefore, it is unknown if the stretch of skin tissues other than from the upper leg can be predicted.

Lastly, to increase variance, more images of skin tissue from more individuals should be included.

Split dataset before image and target transformations

LDS should only be performed on the training set, independently from the validation and test set. This is to prevent leaking data to the training set. By design, the software constructs train, validation and test split datasets with data transformations, including target transformations such as LDS and the Yeo-Johnson transformation. Every split in fact contains all N_{best} images and includes transformations. Just before constructing a dataloader, the datasets are split by index, leaving the dataloaders with non-overlapping data. In future studies, the dataset should be split into subsets with their specific transformations applied. While this increases training fairness, it is expected to decrease performance, as information from the test set is not leaked to the training set.

Moreover, the split indices were generated by shuffling $\{1, 2, 3, \ldots, N_{\text{images}}\}$ and splitting the indices at 80 %, giving indices for the training/validation and test set. Next, a new set of indices, $\{1, 2, 3, \ldots, 80\,\% \cdot N_{\text{images}}\}$ is shuffled and split at 80 %, yielding indices for the training and validation set. All splits were stratified by person, meaning the shuffle was done in

[49]: Lipa et al. (2021), *Does smoking affect your skin?*

[50]: Wilkinson et al. (2020), *Wound healing: cellular mechanisms and pathological outcomes*

such a way that images from the same person could not live in the training or validation set. This way, the training and validation set had overlap with the test set, but not with each other. To create an independent test set, the operation

$$\texttt{actual test} = \texttt{test} - (\texttt{train} \cup \texttt{validation}) \tag{3.15}$$

was performed. However, this does not make use of the full dataset. To achieve that, Equation 3.15 should be rewritten as

$$\texttt{actual test} = \{1, 2, 3, \ldots, N_{\text{images}}\} - (\texttt{train} \cup \texttt{validation}). \tag{3.16}$$

This version of `actual test` has one major drawback, which is that the 20 % highest indices are reserved for the test set, effectively excluding them from the shuffle, leaving a test set with most samples from one person. A future study should perform a train/validation split on shuffled indices after splitting off the test set.

Excluding noise and denoising on stack level

Currently, the stacks are truncated by taking the top N_{best} images of a stack which effectively excludes noisy slices. However, if a full z-stack is noisy, noisy images are still included. These noisy images may still harm training and could be excluded, too. Possibly, excluding noisy stacks can for example be done by calculating the Shannon entropy of all truncated stacks and include stacks with the highest entropy.

In addition to noise exclusion, denoising stacks with three-dimensional Noise2Void [51] or individual slices with Noise2Void2 [52][1] could increase model performance. This is because noise can occlude patterns that describe stretch information.

[51]: Krull et al. (2019), *Noise2Void - Learning Denoising From Single Noisy Images*

[52]: Höck et al. (2023), *N2V2 - Fixing Noise2Void Checkerboard Artifacts With Modified Sampling Strategies And Tweaked Network Architecture*

[1]: At the time of writing, Noise2Void2 is not yet compatible with three-dimensional images.

Three-dimensional and full-size images

The current model relies on single images belonging to stacks. All structural information in the depth direction is disregarded by the neural network. If the neural network is redesigned to recognize patterns in three dimensions, it is expected to better predict the skin stretch properties.

The microscope outputting the SHG images has the ability to image with a resolution of 0.2 µm, which is five times higher than the images used in this study. Higher resolution images contain more detailed information on the collagen structure, and are therefore expected to increase the performance. The presented model has to be redesigned to accept images larger than 258 × 258. Moreover, raw data is stored with a larger dynamic range (16-bit instead of 8-bit). Being able to see small differences in neighboring pixel intensities increases the available information *e.g.* in darker regions, where collagen is sparse, but still significantly present.

Weighting samples by goodness of target fit

The neural network learns from targets that are a result of logistic curves fitted to a series of points. The goodness of fit differs between curves. Fits that do not describe the data well should not negatively impact the model training. One way to achieve this is by re-weighting the loss function as

$$\mathcal{L}_{R^2\,\mathrm{weighted}} = \begin{cases} \mathcal{L}/\left(R^2\right)^b & \text{if } R^2 > a \\ \mathcal{L}/a^b & \text{if } R^2 \leq a, \end{cases} \tag{3.17}$$

where a is a lower bound to R^2 and $b > 0$ can influence the amount of weighting.

3.5.6 Implications

Replacement of mechanical measurements

After training and validating on more data, Skinstression might be used to replace mechanical measurements of stress-strain curves by excising human skin tissue and imaging the dermis layer with an SHG microscope at 0.2 mpp.

3.5.7 Clinical *in vivo* skin studies

An example of clinical use could ultimately be to study skin tissue *in vivo* by a plastic surgeon with a microendoscope as described by Kuzmin et al. [53], but for SHG imaging. Stretch information could be inferred directly from the images made by the microendoscope. This way, the surgeon can acquire specific patient skin information prior to surgery. It is important to note that the model in its current form has low predictive accuracy, and needs further research for this use case.

[53]: Kuzmin et al. (2016), *Third harmonic generation imaging for fast, label-free pathology of human brain tumors*

3.6 Conclusion

The goal of this study was to develop and validate a model that can construct a stress-strain curve corresponding to SHG images of old adult human skin tissue. The convolutional neural network achieved a mean R^2 of −0.36 (SE 0.60) on the test set. Occlusion attribution maps hardly give any explanation to the prediction. Artificially adding and removing collagen seems to respectively increase and decrease the maximum tissue stress, as expected. The model may benefit from more training data and must be validated on a larger test set. Future studies should use nested k-fold cross validation for selecting models and measuring performance. An updated version of the model might replace mechanical skin stretch measurements.

3.7 Supplementary materials

3.7.1 Code

The implementation of Skinstression can be found at siemdejong/shg-strain-stress. The context and container diagram are depicted in Appendix A.4.

4

Developing and validating a clinical context aware multi-instance learning model with self-supervised pre-training on higher harmonic generation images of medulloblastoma and pilocytic astrocytoma in children

Abstract

Background and objective Higher harmonic generation (HHG) microscopy allows for intraoperative feedback. Interpreting intraoperative feedback is time-consuming while time is spare. The time needed for diagnosis might be decreased by artificial intelligence. A clinical context aware multi-instance learning model with self-supervised pre-training (SCLICOM) is developed and validated to automate diagnosis on HHG images.

Methods A five-fold cross-validation study was conducted on HHG data from the Princess Máxima Center for pediatric oncology. Outcomes of interest were pilocytic astrocytoma (PA) and medulloblastoma (MB). A convolutional neural network with and without self-supervised pre-training were validated. A model with clinical context embedding was developed and validated. The performance of the models was assessed by the area under the precision-recall-gain curve (AUPRG) and the mean average precision.

Results HHG biopsy images of 25 children with PA (17) and MB (8) were used. The model achieved a mean average precision of 0.89 (SE 0.05) and 0.41 (SE 0.20) AUPRG. The possibility to select tiles with the highest attentions per tile served useful to help diagnose medulloblastoma or pilocytic astrocytoma.

Discussion SCLICOM showed promising discrimination in predicting PA or MB, but needs further external validation. After additional validation, the updated model may be used to intraoperatively discriminate between pediatric patients with PA or MB or to pre-select interesting regions for diagnosis.

4.1 Introduction

Cancer accounted for about 10 000 000 deaths worldwide in 2020. 246 000 of those were due to brain tumors [54]. In children, brain tumors are the leading cause of cancer mortality. Pediatric brain tumors (PBT) in the Netherlands are treated in the Princess Máxima Center for pediatric oncology (PMC). PMC treats many PBTs, but pilocytic astrocytoma (PA) and medulloblastoma (MB) are two of the most prevalent with an incidence of 0.91 and 0.40 per 100000 children, respectively [55].

[54]: Kocarnik et al. (2022), *Cancer incidence, mortality, years of life lost, years lived with disability, and disability-adjusted life years for 29 cancer groups from 2010 to 2019*

[55]: Adel Fahmideh et al. (2021), *Pediatric brain tumors: Descriptive epidemiology, risk factors, and future directions*

Tumors need to be assessed by pathologists to determine tumor type and severity before making a treatment plan. Starting treatment on time gets increasingly difficult with an expected shortage of pathologists [56]. Pathologists also have access to more health data resulting from multiple imaging techniques, such as magnetic resonance imaging and positron emission tomography, and clinical records than ever before. All this clinical data needs to be processed to optimize patient care. These issues can be addressed by integrating machine learning effectively [57].

[56]: George et al. (2019), *Will I need to move to get my first job?: Geographic relocation and other trends in the pathology job market*

[57]: Parwani (2019), *Next Generation Diagnostic Pathology: Use of digital pathology and artificial intelligence tools to augment a pathological diagnosis*

One of the modalities that can benefit from machine learning is histopathology on whole-slide images (WSI). Biopsies are processed and usually stained with haematoxylin and eosin (HE). HE stained tissues are placed on glass slides and imaged with a microscope to get WSIs. The images are used by pathologists to make a diagnosis. A wide variety of disease patterns can be recognized by histopathology. Pathologists look at large tissue areas and frequently annotate regions of interest to make reasoned diagnoses.

Histopathology images come with artifacts [58] and acquiring them takes a long time and can in general only be done post-operative. Sometimes, intraoperative assessment is desired, as treatment may be tumor type specific. Utilizing higher harmonic generation (HHG) microscopy as a non-invasive and label-free imaging technique enables intraoperative resection feedback.

[58]: Taqi et al. (2018), *A review of artifacts in histopathology*

Manual tumor diagnosis on large HHG images is time-consuming. Various techniques are available to automate tumor diagnosis in WSIs [59]. Blokker et al. [26] has shown that deep learning models trained on THG data can intraoperatively distinguish glioma from epilepsy brain tissue. A convolutional neural network was trained end-to-end in a tile supervised manner, providing the neurosurgeon with a tile-level diagnosis. Another promising branch of techniques relies on multi-instance learning (MIL) where WSIs are cut in tiles. This allows for extraction of interesting features and limiting the amount of data per training batch.

[59]: Litjens et al. (2017), *A survey on deep learning in medical image analysis*

[26]: Blokker et al. (2022), *Fast intraoperative histology-based diagnosis of gliomas with third harmonic generation microscopy and deep learning*

In this work, SCLICOM (from Self-supervised CLInical COntext Multi-instance learning) is proposed. This study includes THG data, as well as SHG, 2PEF, tumor location data, providing an AI with more context to work with. The model is based on DeepSMILE [60] which was originally developed for HE images concerning breast and colorectal cancer. DeepSMILE is a two-stage model that consists of a feature extractor and a MIL classifier. The classifier is extended by including tumor location as clinical context, which is also available to pathologists. The purpose of the product is to intraoperatively classify pilocytic astrocytoma and medulloblastoma using HHG images and clinical context in the form of textual tumor locations. An attention system should give insight into

[60]: Schirris et al. (2022), *DeepSMILE: Contrastive self-supervised pre-training benefits MSI and HRD classification directly from H&E whole-slide images in colorectal and breast cancer*

which areas were relevant for classification. The product may be adapted and trained to account for other tumors or more diseases at once.

4.2 Theory

4.2.1 Medulloblastoma and pilocytic astrocytoma

Pilocytic astrocytoma (PA) is a benign and slow-growing WHO grade I central nervous system (CNS) tumor. PA is mostly characterized by piloid cells, Rosenthal fibers and eosinophilic granular bodies. Often, compact fibrillar areas are next to loose microcystic areas.

Medulloblastoma (MB) is a malignant and fast-growing WHO grade IV CNS tumor. MB is mostly characterized by small and densely packed cells with little cytoplasm.

PA and MB are shown next to normal brain tissue in Figure 4.1. See Spies [61] for more on PA and MB characterization in HHG images.

[61]: Spies (2023), *Validation of higher harmonic generation microscopy for the diagnosis of various pediatric tumors*

4.2.2 Feature extraction

Any input to a neural network can contain a lot of different information. The information is built from a series of (possibly recurring) objects, that may be arbitrarily scattered across the input. These objects determine the meaning of the input.

In the case of images, features exist on multiple abstractions. When completely zooming in, one pixel tells a very local story. A group of pixels may tell the story of a single object. Sets of groups model the interaction between objects. Lastly, combining all sets results in the full image.

Similar to masking, it is useful to extract as much information as possible into a smaller image representation: a feature vector. A feature extractor that creates feature vectors from images can be trained using various algorithms.

One algorithm to train a feature extractor is a simple framework for contrastive learning of visual representations (SimCLR) [62]. SimCLR

[62]: Chen et al. (2020), *A Simple Framework for Contrastive Learning of Visual Representations*

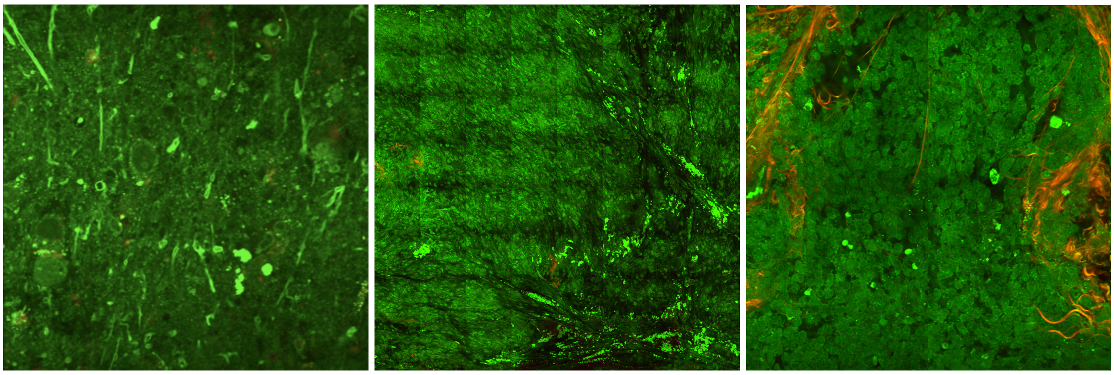


Figure 4.1: Higher harmonic generation images of normal (left), pilocytic astrocytoma (middle), and medulloblastoma (right) tissue. Normal brain tissue contains myelinated axons and neurons. Pilocytic astrocytoma is characterized by piloid cells. Medulloblastoma is characterized by high cellularity.

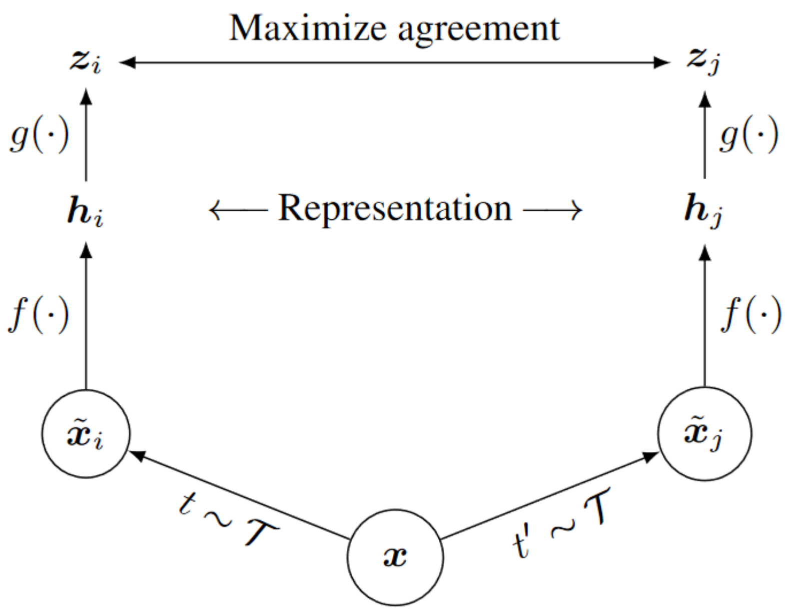


Figure 4.2: A simple framework for contrastive learning of visual representations. Two augmentations of an image are encoded into feature vectors before being projected into a space where a contrastive loss is calculated. Learned feature extractor f encodes images into $\mathbf{h}$ for downstream tasks. Reproduced from Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 'A Simple Framework for Contrastive Learning of Visual Representations'. In: *Proceedings of the 37th International Conference on Machine Learning*. Ed. by Hal Daumé III and Aarti Singh. Vol. 119. Proceedings of Machine Learning Research. PMLR, 13–18 Jul 2020 (Ref. [62]).

learns features by augmenting the same data twice and maximizing the agreement between the representations of those augmentations. No targets are needed. SimCLR is self-supervised.

The original image $\mathbf{x}$ is transformed twice with transform t and $t' \sim T$ to create $\tilde{\mathbf{x}}_{i,j}$. Any transformation can be sampled from transformation space T, but the original authors experiment with cropping, resizing, flipping, color dropping, and color jittering among others. Then, with a chosen convolutional neural network backbone f, the transformed images are encoded in representations $\mathbf{h}_{i,j}$. The representations are then projected with a projection head g to a space in which the loss function between the resulting $\mathbf{z}_{i,j}$ is calculated. The loss function is typically the normalized-temperature cross-entropy loss (NT-Xent) and is defined as

$$\text{NT-Xent} = -\log \frac{\exp(s_{i,j}/\tau)}{\sum_{k=1}^{2N} \mathbb{1}_{[k\neq i]} \exp(s_{i,j}/\tau)}, \tag{4.1}$$

where $s_{i,j}$ is the similarity

$$s_{i,j} = \frac{\mathbf{z}_i \cdot \mathbf{z}_j}{\|\mathbf{z}_i\|\|\mathbf{z}_j\|}, \tag{4.2}$$

N the number of samples in the batch, τ the temperature to scale the similarity with, and $\mathbb{1}_{[k\neq i]} = 1$ if $k \neq i$ and 0 otherwise. The final loss is calculated across all pairs coming from the same parent image in a batch. SimCLR is visualized in Figure 4.2.

A SimCLR trained backbone can then be used as a compression algorithm adapted to the domain it has been trained on.

4.2.3 Multi-instance learning

Now that a SimCLR trained backbone is trained and tiles can be compressed into feature vectors, the features can be used for classification. To

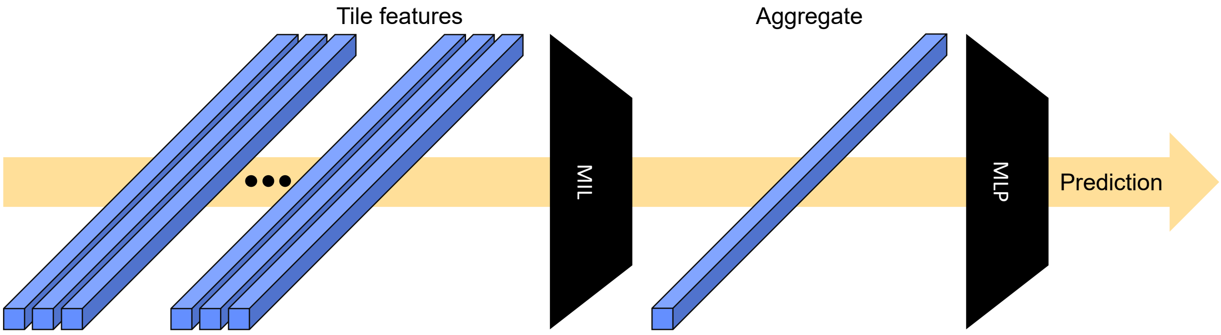


Figure 4.3: Extracted features (tile features in this work) are presented to a multi-layer perceptron (MLP) with learnable weights. The first MLP outputs an aggregate that summarizes all input features. The aggregate is used as input to a classifier MLP with learnable weights that outputs a prediction. Weights are updated based on pre-defined loss between prediction and target.

this end, a classifier must be chosen and trained. Multi-instance learning (MIL) is a suitable technique to deal with multiple features concerning the same outcome.

Classical

Multi-instance learning (MIL) is a supervised learning method. Typically, every instance in a dataset is labelled individually. However, with MIL, the model receives a bag of instances, $B = \{(x_1, y_1), \dots, (x_K, y_K)\}$, where x is an instance with label y. Under the standard assumption, the label of a bag is

$$\begin{aligned} Y(B) &= 1 - \prod_{k}^{K}(1 - y_k) && (4.3) \\ &= \max_k(y_k), && (4.4) \end{aligned}$$

i.e., a bag is positive if at least one instance is positive.

The standard assumption is asymmetric: the meaning of the bag label changes if positive and negative labels are swapped. This assumption might be too restrictive in non-binary problems.

See Figure 4.3 for a schematic overview of a general MIL model.

Attention-based MIL pooling

For pathology studies, it is important to visualize which instances are important for classification. With classical MIL, this is not possible. To overcome the restrictive nature of maximum pooling, Ilse, Tomczak, and Welling [63] propose DeepMIL and use an adaptive weighted average of instances. This weighted average includes learnable weights in an attention-based manner. High weights should be assigned to instances that are likely to have a positive label. The weights allow distinguishing interesting instances from uninteresting ones, see Figure 4.4. The attention weights of specific instances (*e.g.*, patches in an image) explain how the model comes to its diagnosis prediction which could be compared with the doctor's diagnosis.

[63]: Ilse et al. (2018), *Attention-based Deep Multiple Instance Learning*

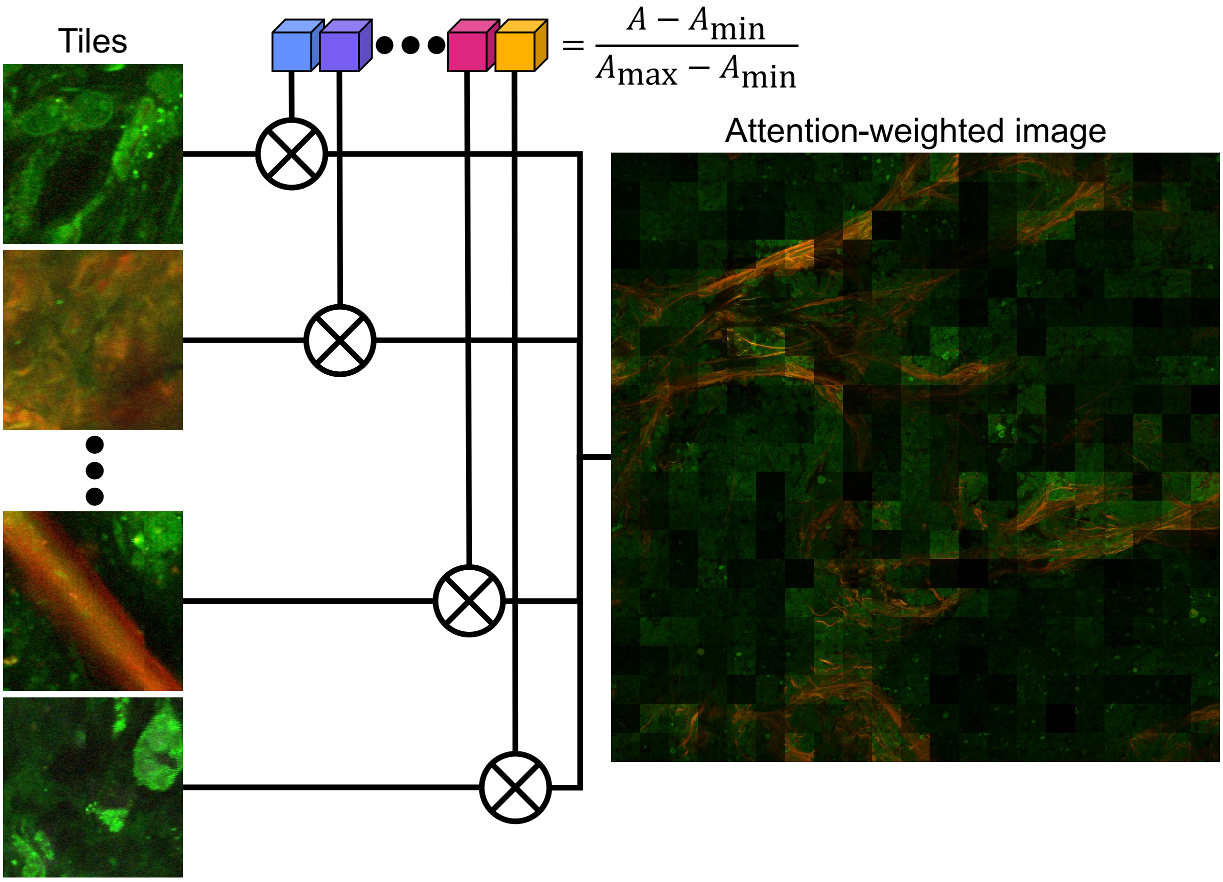


Figure 4.4: Visualizing tile importances. The attention weights resulting from VarMIL are min-max-normalized and multiplied with their corresponding tile. The output is an attention weighted image with bright parts relating to high attention and dark parts relating to low attention. Note that dark tiles can still have high attentions if the original image contains dark patches with useful information.

Let $H = \{\mathbf{h}_1, \ldots, \mathbf{h}_K\}$ be a bag of K embeddings. The weighted average of H is

$$\mathbf{z} = \sum_{k=1}^{K} a_k \mathbf{h}_k, \tag{4.5}$$

where

$$a_k = \frac{\exp\left[\mathbf{w}^T \tanh(\mathbf{V}\mathbf{h}_k^T)\right]}{\sum_{j=1}^{K} \exp\left[\mathbf{w}^T \tanh(\mathbf{V}\mathbf{h}_k^T)\right]}, \tag{4.6}$$

where $\mathbf{w} \in \mathbb{R}^{L\times 1}$ and $\mathbf{V} \in \mathbb{R}^{L\times M}$ are the learnable parameters. The denominator ensures the weights sum to 1. $\mathbf{z}$ is further processed in an MLP for classification. The weights can also be multiplied by the corresponding input tiles to show which features were important for the prediction.

Variance MIL pooling

DeepMIL has the disadvantage of discarding any inter-tile information. In a clinical setting, this high-level information can model *e.g.* the intratumor heterogeneity or tumor border shape.

Schirris et al. [60] propose Variance MIL (VarMIL) which adds a learnable

[60]: Schirris et al. (2022), *DeepSMILE: Contrastive self-supervised pre-training benefits MSI and HRD classification directly from H&E whole-slide images in colorectal and breast cancer*

attention-weighted variance,

$$\boldsymbol{\sigma} = \frac{K}{K-1} \sum_{k=1}^{K} a_k \left(\mathbf{h}_k - \mathbf{z}\right)^2 , \tag{4.7}$$

to capture global features. The weighted average and variance are concatenated in a single vector, such that

$$\hat{\mathbf{z}} = \begin{pmatrix} \mathbf{z} \\ \boldsymbol{\sigma} \end{pmatrix} . \tag{4.8}$$

As with DeepMIL, the weights can still be used to highlight tiles.

Clinical Context MIL

Clinical contexts such as locations of tumor resections are important for clinical decisions by pathologists. For example, medulloblastoma is mostly found in the fourth ventricle or cerebellar parenchyma [64]. This information along with the attention weighted tiles (see Subsection 4.2.3) may lead to better performance. As pathologists also have access to this information, it is reasonable for an AI model to use the same available information.

[64]: Millard et al. (2016), *Medulloblastoma*

In this work, Clinical Context MIL (CCMIL) is proposed. CCMIL builds on VarMIL, with the addition of clinical information as input to the classification layer. The clinical information is presented as a string of characters, a sentence, which can be anything clinically relevant. The sentence is condensed to an $H \times 1$-dimensional text embedding. The embedding is concatenated to the output of the MIL aggregate, such that

$$\tilde{\mathbf{z}} = \begin{pmatrix} \hat{\mathbf{z}} \\ \boldsymbol{\mathscr{C}} \end{pmatrix} , \tag{4.9}$$

where $\boldsymbol{\mathscr{C}}$ is the clinical context text embedding. $\tilde{\mathbf{z}}$ is used as input for the trainable classifier.

The text embedding can be created in various ways[1]. The most direct way would be to create a list of possible sentences and convert the input text to a one-hot encoded vector, effectively selecting a specific sentence from the vocabulary. The drawback of this method is that it cannot handle out-of-vocabulary (OOV) words, which requires an extensive vocabulary that might become obsolete.

One of the most promising methods for creating text embeddings for use in downstream tasks are transformer based methods [66]. Attention-based models such as BERT [67] and ELMo [68] are able to distinguish important words and can distil the right meaning of homographs. With BERT's `[CLS]`-token, a sentence classification can be created to embed a sentence to a point in a high-dimensional space. These models can learn OOV words which enables them to be used in new contexts.

[66]: Vaswani et al. (2017), *Attention is All you Need*

[67]: Devlin et al. (2019), *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*

[68]: Peters et al. (2018), *Deep Contextualized Word Representations*

The aggregator of CCMIL is visualized in Figure 4.5.

[1] See Khattak et al. [65] for a review.

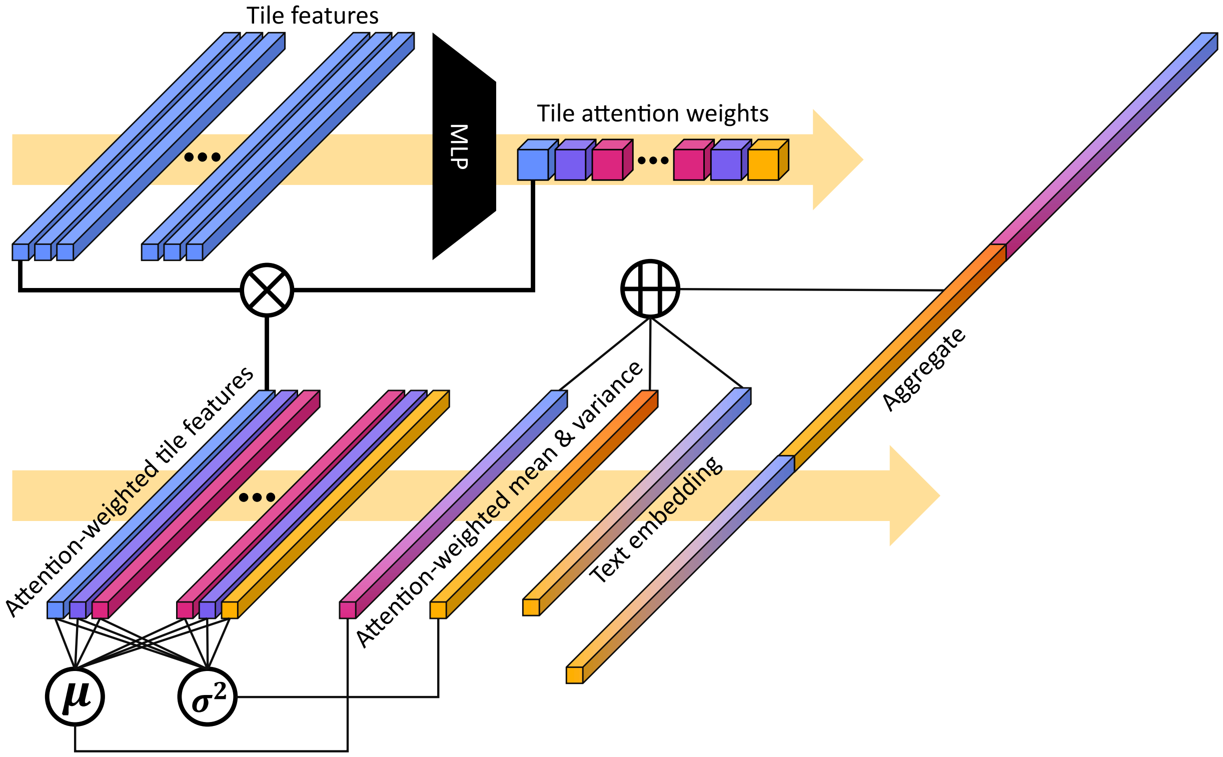


Figure 4.5: Clinical Context Multi-Instance Learning (CCMIL) aggregator. Just like in Variance MIL (VarMIL), tile features are presented to a multi-layer perceptron (MLP) to learn attention weights per tile. The attention weights are multiplied with their corresponding tile features to get attention-weighted tile features. From these, the mean μ and variance σ^2 are calculated and concatenated. Extending VarMIL, CCMIL further concatenates a text embedding that may contain clinical context of the input image. The aggregate is further processed in another MLP for classification.

4.2.4 Classification performance metrics

Receiver operating characteristic curve

To compare the performance of different models, it is common practice to compare their receiver operating characteristic (ROC) curve. The true positive rate, $TPR = TP/(TP + FN)$, is plotted against the false positive rate, $FPR = FP/(FP + TN)$, at different thresholds for a prediction score to be counted as positive or negative. The ROC curve always starts at (0, 0) and ends at (1, 1). The diagonal shows the values above which a classifier performs better than a classifier with an accuracy of 0.5, where accuracy is

TP: True Positive
FN: False Negative
FP: False Positive
TN: True Negative

$$ACC = \frac{TP + TN}{TP + TN + FP + FN}. \tag{4.10}$$

This, however, is not the baseline to beat in the case of an imbalanced dataset. The always positive classifier has an accuracy that depends on the fraction of positive examples in the dataset π. The corresponding baseline is determined by the function

$$TPR = \frac{1-\pi}{\pi} FPR + 2 - \frac{1}{\pi}, \tag{4.11}$$

see Appendix B.1.1 for a derivation. A perfect classifier has an ROC curve that goes from (0, 0) to (0, 1) to (1, 1), such that the area under the ROC curve (AUROC) is the highest.

Precision-Recall curve

The ROC curve is not well-suited for imbalanced datasets [69]. Another way to visualize model performance is plotting precision, prec = $TP/(TP + FP)$, against TPR (recall, rec), at different thresholds for a prediction score. The Precision-Recall (PR) curve ignores true negatives and always starts at (0, 1) and ends at (1, 0). The random classifier has its baseline at prec = π. The baseline of the always positive classifier is the point where rec = 1 and prec = π.

[69]: Saito et al. (2015), *The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets*

Precision and recall can be combined into the F_1-score. The F_1-score is the harmonic mean of the precision and recall:

$$F_1 = \frac{2}{\text{rec}^{-1} + \text{prec}^{-1}} = \frac{2TP}{2TP + FP + FN}. \tag{4.12}$$

In general, F_β is a weighted harmonic mean where $\beta \in \mathbb{R}+$, such that recall is β times as important as precision:

$$F_\beta = (1 + \beta^2)\frac{\text{prec} \cdot \text{rec}}{(\beta^2 \cdot \text{prec} + \text{rec})}. \tag{4.13}$$

The F-score can take values from 0 (if precision or recall is 0) to 1 (for perfect precision and recall).

The iso-F_1 curve with the F_1-score of the always positive classifier can be seen as the baseline to beat, and is defined as

$$\text{prec}_{\text{always positive classifier}} = \frac{F_{1,\text{ always positive classifier}} \cdot \text{rec}}{2 \cdot \text{rec} - F_{1,\text{ always positive classifier}}}, \tag{4.14}$$

see Appendix B.1.2 for a derivation.

A perfect classifier has a PR curve that goes from (0, 1) to (1, 1), to (1, 0), such that the area under the PR curve (AUPR) is the highest.

Precision-Recall-Gain curve

The PR curve has some major problems [70]. Firstly, all baselines are non-universal. They depend on the distribution of positives. Secondly, the arithmetic mean of the F_1 scores are often reported, while this is not meaningful. It is only meaningful to take the harmonic mean.

[70]: Flach et al. (2015), *Precision-Recall-Gain Curves: PR Analysis Done Right*

For more considerations, see Flach and Kull [70], which proposes the Precision-Recall-Gain (PRG) curve. This curve is a harmonic transformation of the PR curve, such that the random baseline is at prec-Gain = $PG = 0$ for all rec-Gain = RG and the always positive classifier is at (1, 0). The transformed F_β-score, the FG_β-score is defined as

[70]: Flach et al. (2015), *Precision-Recall-Gain Curves: PR Analysis Done Right*

$$FG_\beta = 1 - \frac{1 - \pi}{\pi}\frac{FP + \beta^2 FN}{(1 + \beta^2)TP}. \tag{4.15}$$

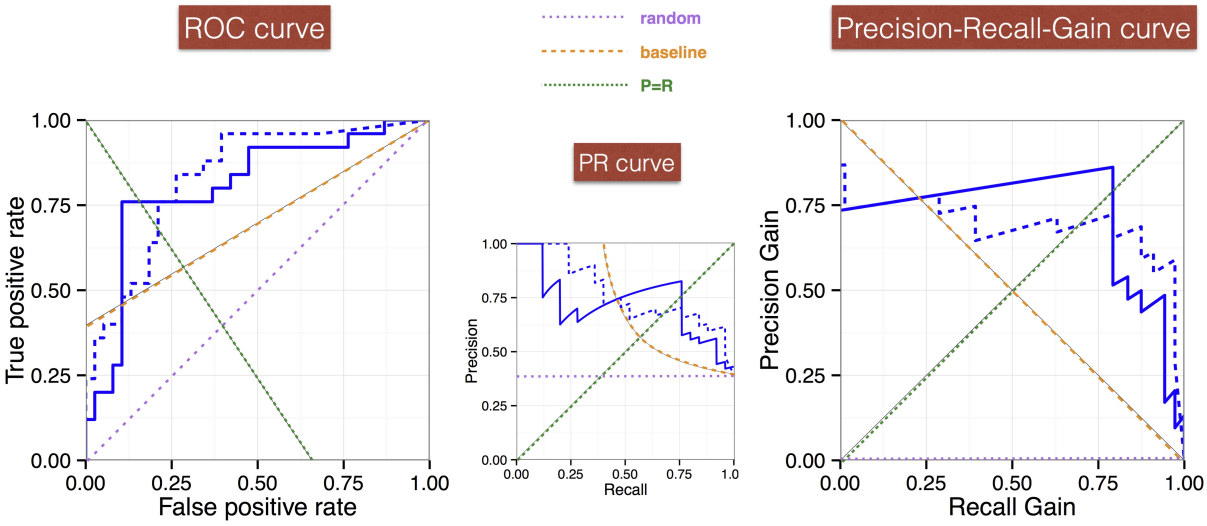


Figure 4.6: Receiver operating characteristic (ROC) curve, Precision-Recall (PR) curve, and Precision-Recall-Gain curve of two models. The random (dotted, purple) and always positive classifier (dashed, yellow) baselines are shown, as well as the precision = recall lines. The area under the PR curve wrongly suggests that the model with the dashed line performs better, while the area under the PRG curve suggests it's the worst. Reproduced from Peter Flach and Meelis Kull. 'Precision-Recall-Gain Curves: PR Analysis Done Right'. In: *Advances in Neural Information Processing Systems*. Ed. by C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett. Vol. 28. Curran Associates, Inc., 2015 (Ref. [70]).

The harmonic transformation ensures that iso-FG_1 lines are linear, with function

$$PG = -RG + 2 \cdot FG_1, \tag{4.16}$$

such that the minor diagonal is always the baseline to beat.

A perfect classifier has a PRG curve that looks akin to the one in a PR curve, such that the area under the PRG curve (AUPRG) is the highest.

A summary of the ROC, PR, and PRG curve is shown in Figure 4.6. This shows that the PRG curve is better suited to compare models as opposed to the PR curve.

4.2.5 Intersection over union

Image segmentations can be assessed by calculating the intersection over union (IoU) between the segmentation A and a predefined ground truth B, as

$$\text{IoU}(A, B) = \frac{|A \cap B|}{|A \cup B|}, \tag{4.17}$$

where $0 \leq \text{IoU} \leq 1$. If the segmentation does not intersect the ground truth, then $\text{IoU} = 0$.

4.2.6 Masking

Large pathology images do not only include tissue. They also include empty space, mostly near the border, but possibly also within the tissue, *e.g.*, in the case of air bubbles. As there is no information in the empty space, it is useless for an AI. A model converges faster if the non-informational parts can be skipped. Skipping over these areas can be achieved with masking. In this study, three masking algorithms designed for pathology are considered.

FESI

Foreground Extraction from Structure Information (FESI) [71] is an algorithm that relies on the distance transform of a Laplacian transformed grayscale image and flood filling from the point with the highest distance. Improved FESI [72] is an improvement of FESI where the input image is change to LAB color space and the L and A channels are changed to maximum intensity. It further uses a Gaussian filter instead of the absolute value of the Laplacian.

[71]: Bug et al. (2015), *Foreground extraction for histopathological whole slide imaging*

[72]: Riasatian et al. (2020), *A Comparative Study of U-Net Topologies for Background Removal in Histopathology Images*

EntropyMasker

Another way to use structure information in pathology images is by looking at the entropy profile as done in EntropyMasker [73]. EntropyMasker works by converting the input image to grayscale and calculating the local entropy. Then, the entropy is binned, and a threshold is determined from the minimum in the histogram. Lastly, the threshold is applied to the entropy image to end up with a binary mask.

[73]: Song et al. (2023), *An automatic entropy method to efficiently mask histology whole-slide images*

4.3 Methods

4.3.1 Source of data

Data is collected from 26 September 2022 until 13 April 2023 during routine care in the Princess Máxima Center for pediatric oncology. HHG images were acquired before HE-acquisition as described in [61]. The dataset is used for train, validation, and test subsets.

[61]: Spies (2023), *Validation of higher harmonic generation microscopy for the diagnosis of various pediatric tumors*

4.3.2 Participants

The data is acquired in the Princess Máxima Center for pediatric oncology, Utrecht. Images were made of brain and solid tumor tissues excised from children (0–16 yr). All patients were eligible for imaging regardless of previously received therapy.

4.3.3 Data preparation

The target classification is transformed to unique numbers.

Images are exported from the raw HHG microscope data by Spies [61]. Overview images of 1 mpp and close-up images of 0.2 mpp of histologically interesting areas were made. To be able to use both images, the overview images were scaled to the same resolution as close-up images using Lanczos interpolation.

[61]: Spies (2023), *Validation of higher harmonic generation microscopy for the diagnosis of various pediatric tumors*

Given the size of the images (200 MB and 7700 × 7900 8-bit RGB pixels on average) and to be able to use MIL, the images are subdivided into non-overlapping tiles of 224 × 224. Tiles overflowing the image are skipped.

Many overview images contain a large empty space without tissue. Tiles in these regions do not contain any useful information. Ideally, empty areas are masked such that only informative tiles are included. For this, multiple algorithms specifically written for H&E data are evaluated on HHG data (see Subsection 4.3.4). The best performing algorithm, EntropyMasker, is chosen to extract the foreground.

Rescaling, tiling and masking has been done with a fork [74] of the Deep Learning Utilities for Pathology project [75] that includes a reimplementation of the EntropyMasker.

[74]: de Jong et al. (2023), *Deep Learning Utilities for Pathology, branch entropy_masker*

[75]: Netherlands Cancer Institute et al. (2023), *Deep Learning Utilities for Pathology*

4.3.4 Masking

Three different masking algorithm designed for HE pathology are evaluated against 8 masks manually made from overview images with QuPath [76]. The algorithms include FESI, improved FESI, and EntropyMasker. For EntropyMasker, the local entropy was calculated with a disk with a radius of 5 px as structure element. For every mask, the IoU is calculated and its mean is reported.

[76]: Bankhead et al. (2017), *QuPath: Open source software for digital pathology image analysis*

4.3.5 Outcome

The model predicts which tumor type is mostly present in the input image within seconds. The tumor type prediction is assessed by consensus of at least two pathologists, based on HE whole-slide images, tumor location, and patient demographics.

The model also outputs an attention map explaining which tiles were important for the prediction.

4.3.6 Predictors

With the ultimate goal of predicting specific tumor types, an obvious way of choosing predictors is to directly use diagnoses made by pathologists without choosing an intermediate predictor. All available diagnoses for HHG data are summarized in Table 4.2. As pilocytic astrocytoma and medulloblastoma were predominantly imaged, they are chosen as direct predictors. Although it was originally the goal to also distinguish ependymoma from pilocytic astrocytoma and medulloblastoma, there were not enough available samples to train on.

4.3.7 Models

This section shows the methods to obtain a model trained on HHG images, diagnoses and clinical context. The final model, a combination of two models together named SCLICOM (from Self-supervised pretraining and CLInical COntext-aware Multi-instance learning), should be able to explain itself via attention maps. The pipeline is summarized in Figure 4.7.

The model consists of two stages: the feature extractor and the classifier. They are trained separately.

Feature extractor

The feature extractor is a convolutional neural network with a ShuffleNetV2 (x1.0) [77] backbone, provided by imgclsmob [78], outputting a vector with length 1024. Two backbones were used as feature extractor. One was pre-trained on ImageNet and another feature extractor has an He initialized ShuffleNetV2 backbone and was trained using SimCLR. The SimCLR projection head consisted of two linear layers with dimensions 1024 and 128.

[77]: Ma et al. (2018), *ShuffleNet V2: Practical Guidelines for Efficient CNN Architecture Design*

[78]: contributors (2023), *imgclsmob*

Classifier

The classifier is a MIL model applied to the features extracted by the pretrained feature extractor. DeepMIL, VarMIL and CCMIL are used as classifiers. For DeepMIL, the attention block consists of a linear layer of size 256 with dropout and a tanh activation function and another linear layer with dropout, resulting in a scalar. The classifier block consists of a linear layer with dropout and softmax activation function and has an output size of the number of classes to predict, *i. e.* two. For VarMIL, the last linear layer accepts an input of twice the size, to account for the variance vector. For CCMIL, the last linear layer maps textual tumor locations (*e. g.* "fourth ventricle") to a 312-dimensional space. Text is embedded with TinyClinicalBERT [79], provided by Huggingface Transformers [80], using its `[CLS]`-token.[2] During training, BERT's parameters were frozen. All other parameters were He initialized. To validate BERT, the location embeddings were projected to two dimensions with t-SNE.

[79]: Rohanian et al. (2023), *Lightweight Transformers for Clinical Natural Language Processing*

[80]: Wolf et al. (2020), *Transformers: State-of-the-Art Natural Language Processing*

[2] TinyClinicalBERT is trained on the critical care MIMIC-III database [81]

Internal validation

The data was split in five training and test data folds, stratified by case. All training splits were further split randomly, again stratified by case. The flow of images to splits is visualized in Appendix B.2.

Pre-training

The latter model is trained on 1 NVIDIA A30 GPU for 2 days with a bag size of 256 with gradients accumulated over two epochs to imitate larger bags. The Adam optimizer was used with a learning rate of 3×10^{-4} and $\beta_{1,2} = \{0.9, 0.999\}$ without weight decay and without a learning rate scheduler.

The pretrained models are internally assessed by visualizing the extracted features in two ways. First, tiles corresponding to ten nearest neighbors in feature space are compared. Second, the features are projected with two-dimensional t-SNE after extracting the ten principal components

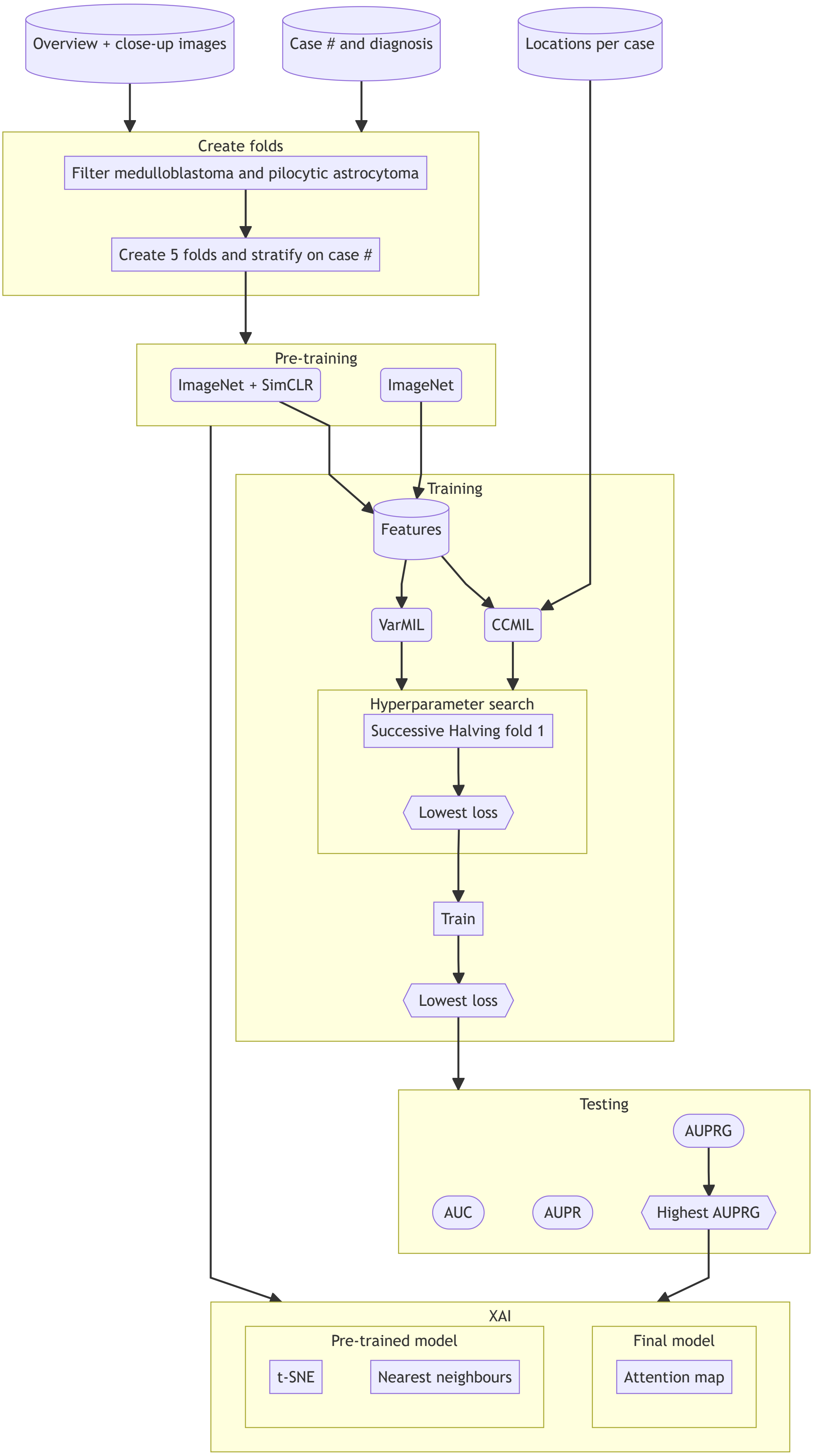


Figure 4.7: Flowchart for training of Self-supervised pre-training and CLInical COntext-aware Multi-instance learning (SCLICOM). The images are filtered by diagnosis and divided in five folds. An ImageNet initialized SimCLR model is trained. From this model and an ImageNet initialized model, features are extracted. The extracted features are used to train VarMIL and CCMIL models. One fold for every model is used to search for hyperparameters with which the models are further trained. The models are tested on the test set and AUC, AUPR, and AUPRG are reported. Attention maps are made from the best performing model.

with PCA. The t-SNE projections are colored by image and case identifier, and diagnosis.

Training

In total, 15 models are trained using PyTorch Lightning [82]: 3 model definitions on 5 folds. For every fold and model a Successive Halving hyperparameter search is performed on 1 NVIDIA A100 GPU partitioned in 8 for a minimum of 30 epochs and a maximum of 500 epochs with resources managed by Ray Tune [83] and 100 trials suggested by a grouped multivariate TPE sampler provided by Optuna [44]. See Table 4.1 for the hyperparameter search space. Using the configuration that lead to the model with the lowest validation loss is used for training on all splits for 2000 epochs. The models were trained on 1 NVIDIA A100 GPU for 2 hours with a bag size of 1, *i.e.* all tiles from one image at a time. The learning rate was varied with the cosine annealing scheduler.

[82]: Falcon et al. (2019), *PyTorch Lightning*

[83]: Liaw et al. (2018), *Tune: A Research Platform for Distributed Model Selection and Training*

[44]: Akiba et al. (2019), *Optuna: A Next-Generation Hyperparameter Optimization Framework*

Table 4.1: Hyperparameter search space with dropout, learning rate, momentum and weight decay.

Parameter	Min.	Max.
Dropout	0	1
Learning rate	10^{-5}	10^{-2}
Momentum	0	1
Weight decay	10^{-4}	1

Testing

All models from the same initialization were evaluated against a test set. Performance is visualized using an ROC, PR and PRG curve. AUC, AUPR, and AUPRG with their 95 % confidence interval are reported to compare models. Metrics were calculated by Torchmetrics [84] and pyprg [70].

[84]: Nicki Skafte Detlefsen et al. (2022), *TorchMetrics - Measuring Reproducibility in PyTorch*

[70]: Flach et al. (2015), *Precision-Recall-Gain Curves: PR Analysis Done Right*

Attention maps

To verify and to gain knowledge of the inner workings of the model, attention maps are created. The attention maps are acquired by multiplying the min-max-normalized attention weights with their corresponding tiles. The maps are compared with tumor annotations from a pathologist. The IoU of the tumor is reported.

4.4 Results

4.4.1 Participants

In total, there were 45 available cases. Most cases were concerned with pilocytic astrocytoma (17) and medulloblastoma (8). This amounts to a total of 134 images of which are 26 overview images and 108 close-up images. These cases were included in the training and testing of the model. The included images were taken between 26 September 2022 and 13 April 2023. No participants have received prior treatment and no images of follow-up treatments are included.

The age and sex of the subjects per fold are summarized in Figure 4.8.

4.4.2 Masking

Diagnosis	Count
Pilocytic astrocytoma	**17**
Medulloblastoma	**8**
Craniopharyngioma	5
Ganglioglioma	3
Ependymoma	1
Glioma	1
Medullomyoblastoma	1
Diffuse midline glioma	1
Dysembryoplastic neuroepithelial tumor	1
Pituitary Neuroendocrine Tumors	1
Atypical choroid plexus papilloma	1
Neuroendocrine tumor	1
Infantile hemispheric glioma	1
Subependymal giant cell astrocytoma	1
Reactive	1
No neoplasm	1

Table 4.2: Number of cases per diagnosis. Bold rows were included in this study.

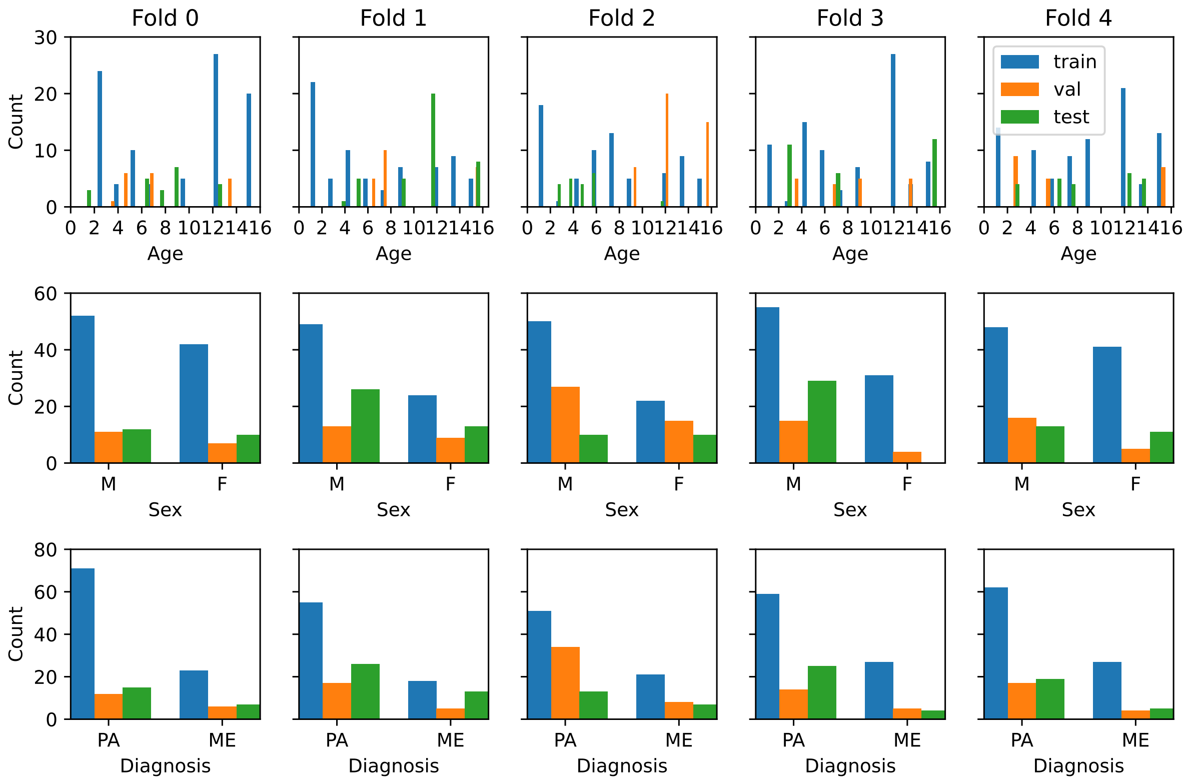


Figure 4.8: Age, sex, and diagnosis distribution per training, validation and test split for every fold. Only pilocytic astrocytoma (PA) and medulloblastoma (ME) cases are included.

FESI, Improved FESI, and EntropyMasker have been evaluated against 8 manually masked HHG overview images. Figure 4.9 shows the binary masks created by the algorithms. The mean IoU for every algorithm is reported in Table 4.3. EntropyMasker masks HHG overview images significantly better than (Improved) FESI.

Table 4.3: Intersection over Union (IoU) for three pathology masking algorithms applied to HHG overview images. **Bold** indicates statistically significant ($p < 0.01$) greater value than second-greatest score.

Algorithm	IoU (SE)
FESI	0.64(3)
Improved FESI	0.64(6)
EntropyMasker	**0.92(2)**

4.4.3 Model specification

The models are combination of a SimCLR pretrained feature extractor and a MIL classifier. For the MIL classifier, DeepMIL, VarMIL, and CCMIL were trained. Model weights for the feature extractor and classifier per fold can be downloaded from siemdejong/sclicom.

4.4.4 Performance

Feature extractor

Nearest neighbors in image space The nearest neighbors are calculated in feature space after which the ten corresponding nearest neighbors of a random tile are shown. The nearest neighbors are calculated for a SimCLR pretrained backbone as well as a backbone pretrained on ImageNet. The results are shown in Figure 4.10.

T-SNE feature embedding To further see how the features are distributed, the higher dimensional feature vectors of the validation set of fold 0 are projected onto two-dimensions using t-SNE at a perplexity of 100. The t-SNE embeddings are shown three times with different colors for diagnosis, case and image in Figure 4.11.

Classifier

The cross entropy loss for all folds for the SimCLR + CCMIL classifier training is shown in Figure 4.12. Loss curves for the VarMIL models are similar. Note the validation loss of fold 1 diverging from the corresponding training loss, indicating overfitting.

The ROC, PR, and PRG curves of the model applied on the test set are shown in Figure 4.13. The mean AUC, AUPR, and AUPRG are summarized in Table 4.4.

When making conclusions on performance based on PR curves, one might find that fold 1 and 2 of the SimCLR + VarMIL model perform comparably. Eliminating data outcome bias with the PRG curve shows a significant difference between those models where fold 1 is performing worse than the always positive classifiers. Based on the ROC curve, one might conclude that the model of fold 2 only performs marginally better than that of fold 1.

The performance of fold 1 is consistently much lower than the other folds, while fold 0 and 2 are consistently higher. One explanation could be that by coincidence the data quality is lower in fold 1 and higher in fold 0

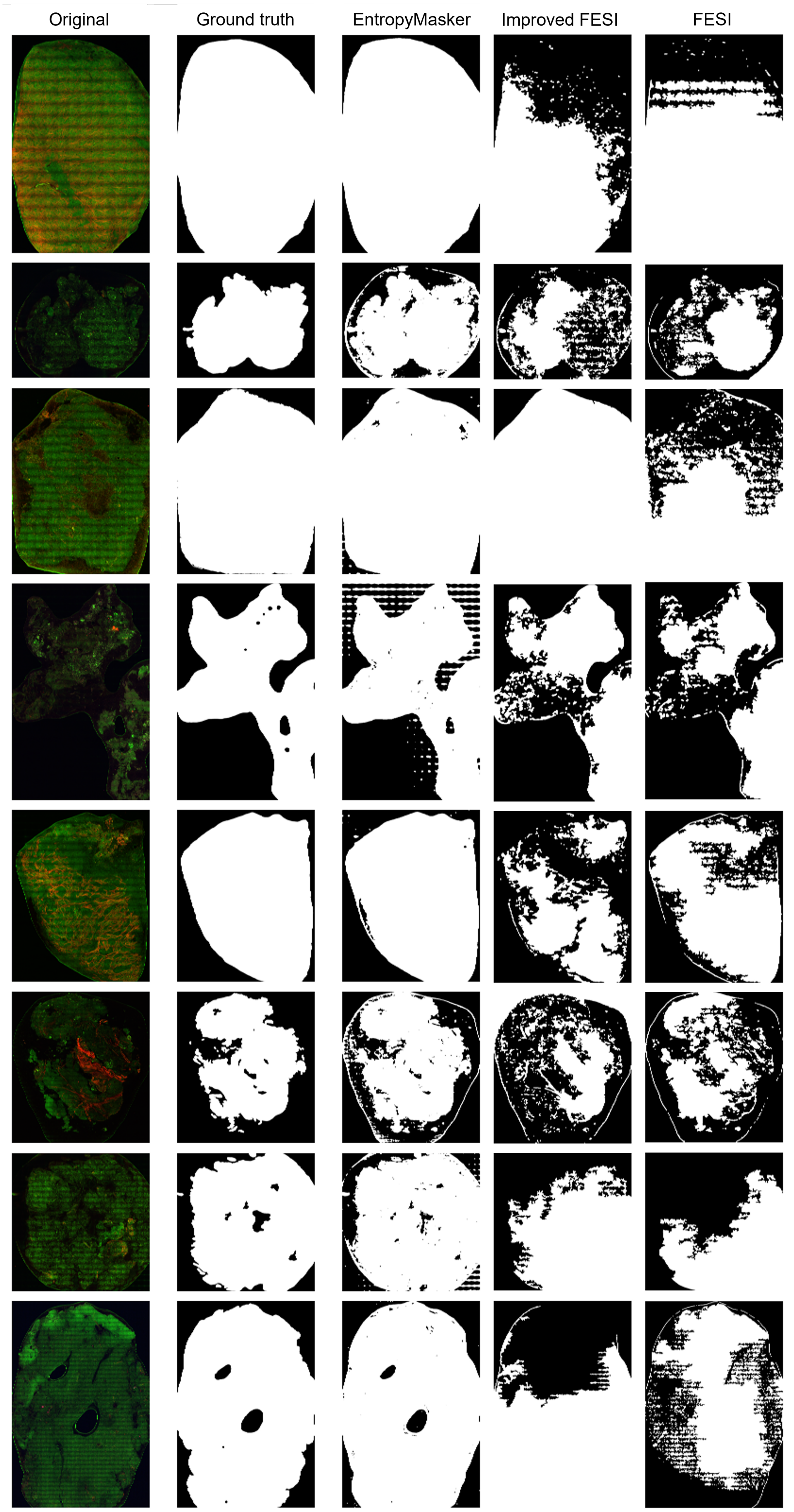


Figure 4.9: Masks generated by EntropyMasker, Improved FESI, and FESI on HHG overview images. The first two columns show the original images and the ground truth segmentation. The other columns show the generated masks.

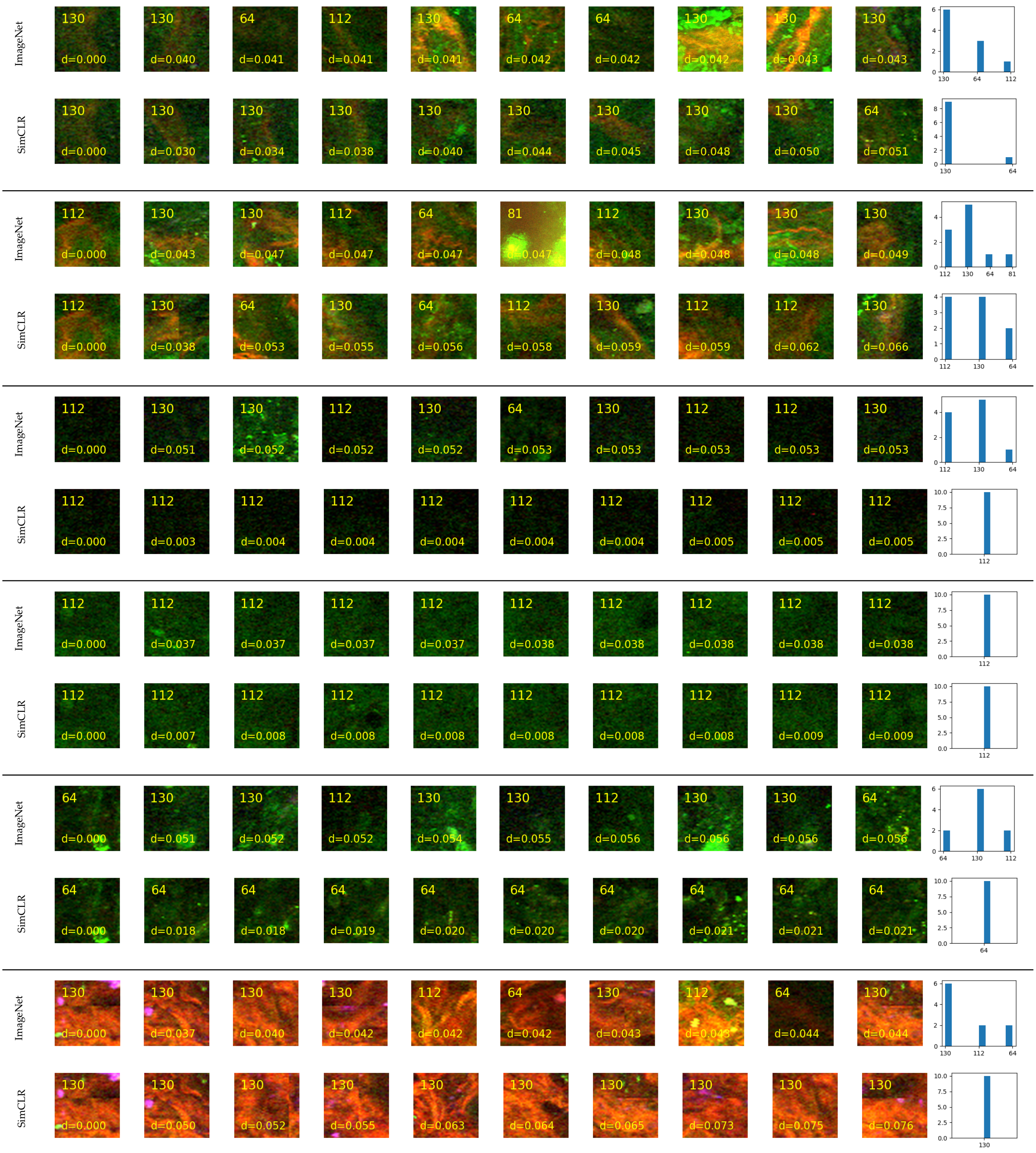


Figure 4.10: Nearest neighbors visualization in image space. Neighbors were first calculated in feature space and their corresponding tiles are plotted. Every first row after dividers are calculated using the ImageNet pretrained backbone. Every second row is calculated using the SimCLR pretrained backbone. Bottom left shows the Euclidean distance (d) with respect to the target image in the first column. The last column shows the distribution of cases for the first 10 nearest neighbors.

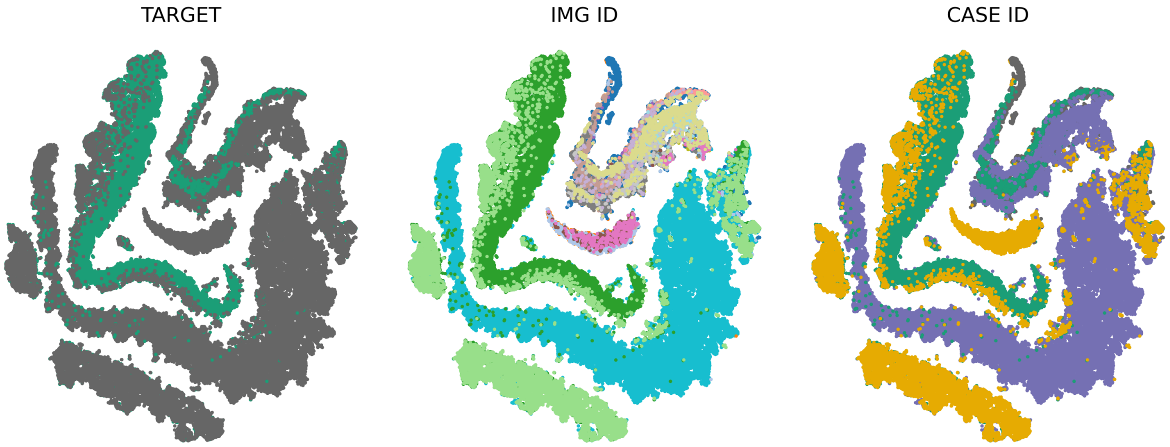


Figure 4.11: Features of the validation set of fold 0 extracted by a SimCLR pretrained ShuffleNetV2 network visualized in two-dimensional t-SNE projections at a perplexity of 100. Points are colored by diagnosis, case and image. Colors between subplots do not necessarily correlate.

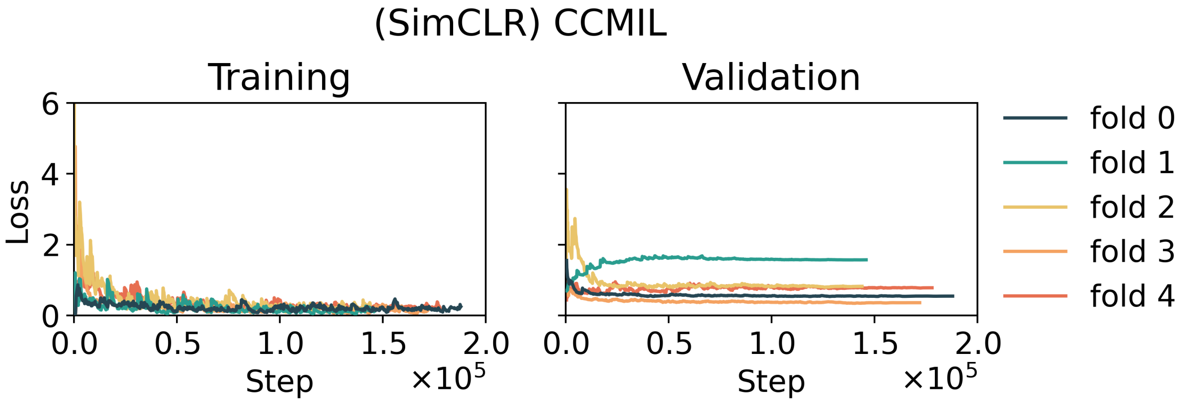


Figure 4.12: Cross entropy loss for training the classifier of the SimCLR + CCMIL model. The loss for the training and validation all five folds are shown.

and 2. To investigate this, the entropy in the upper part of the power spectrum, and the kurtosis are measured and shown in Figure 4.14. Fold 1 shows a wide spread in kurtosis and entropy The Pearson correlation coefficient (corr) between the test AUPRG and entropy (h) and kurtosis (k) mean are calculated, like

$$\text{correlation}_h = \mathbb{E}\left\{[\text{AUPRG} - \mathbb{E}\left(\text{AUPRG}\right)][H - \mathbb{E}\left(H\right)]\right\}, \qquad (4.18)$$

$$\text{correlation}_k = \mathbb{E}\left\{[\text{AUPRG} - \mathbb{E}\left(\text{AUPRG}\right)][K - \mathbb{E}\left(K\right)]\right\}. \qquad (4.19)$$

Entropy and kurtosis have a small correlation with test AUPRG of 0.18 and -0.14, respectively.

4.4.5 Explainability

Attention weighted images

Attention weighted images are shown in Figure 4.15. They are created from medulloblastoma and pilocytic astrocytoma data from the test set of fold 0 using the corresponding model. Only a small portion of the tiles

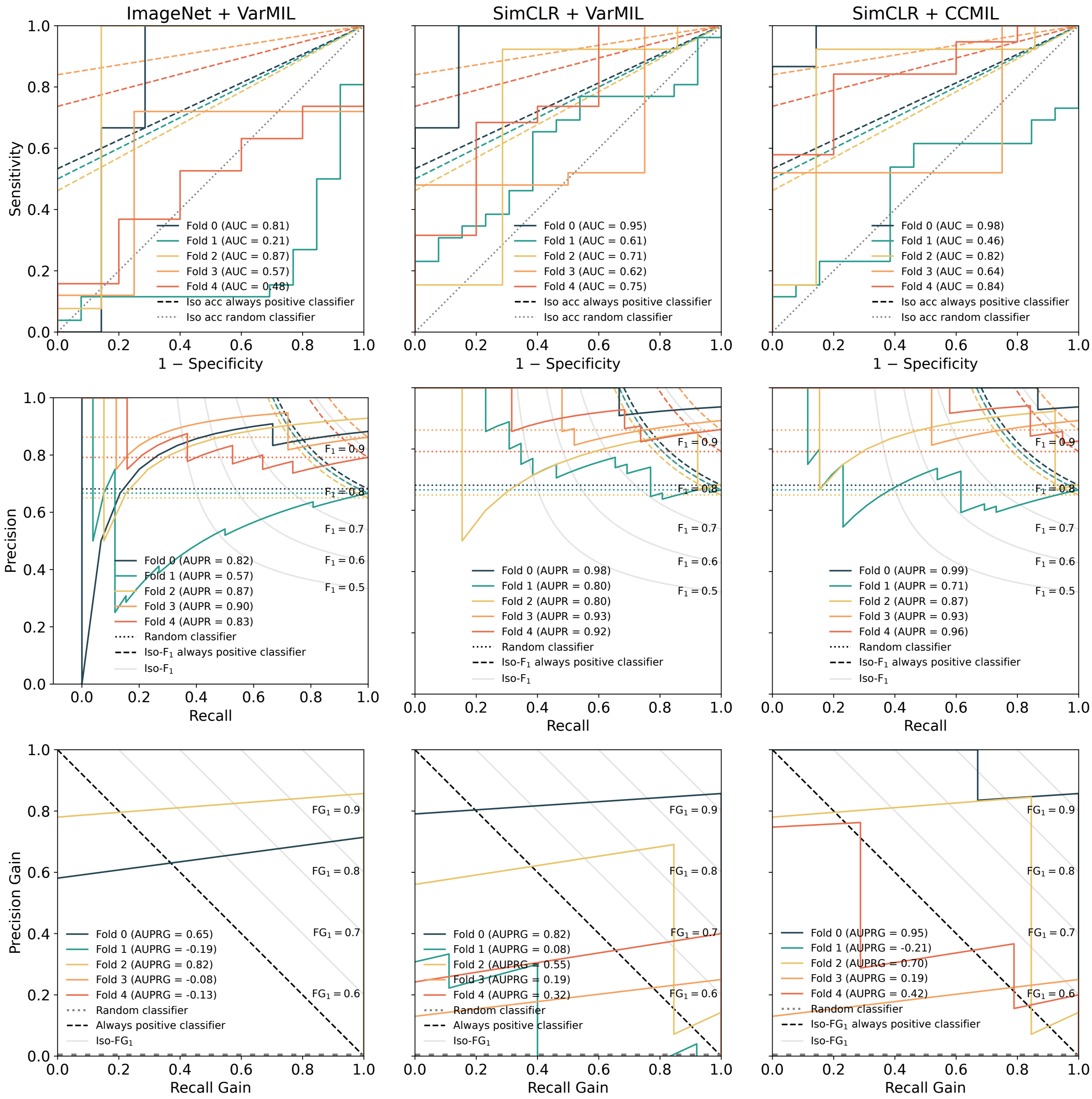


Figure 4.13: ROC, PR, and PRG curves of the model applied on the test set. Top row: ROC curves (color solid) for *feature extractor + classifier* with iso-accuracy lines of the always positive classifier (color dashed) and a random classifier (dotted) are shown. The iso-accuracies all go through a classifier with sensitivity = 1 − specificity = 1, but are non-universal. The iso-accuracy of a random classifier is universal. For every fold, the AUC is reported. Middle row: PR curves (color solid) with iso-F_1 lines of the always positive classifier (color dashed) and a random classifier (color dotted). The iso-F_1 curves go to the upper right corner for increasing F_1. The iso-F_1 lines of the random classifiers go to 1 for increasing bias to pilocytic astrocytoma examples. Iso-F_1 lines are not universal. For every fold, the AUPR is reported. Bottom row: PRG curves (color solid) with iso-FG_1 lines (solid gray). The precision gain of the random classifier (dotted on precision gain = 0) and the always positive classifier baseline (dashed) are universal. Points on the PRG curves with precision gain < 0 or recall gain < 0 are not shown. For every fold, the AUPRG is reported.

Table 4.4: AUC, AUPR, and AUPRG with standard error of three different models applied to the validation and test set. No value is statistically significant ($p < 0.05$).

Model	Validation			Test		
	AUC	AUPR	AUPRG	AUC	AUPR	AUPRG
ImageNet + VarMIL	0.86(5)	0.96(2)	0.57(16)	0.59(12)	0.80(6)	0.21(22)
SimCLR + VarMIL	0.69(9)	0.89(3)	0.28(13)	0.73(6)	0.88(4)	0.39(13)
SimCLR + CCMIL	0.71(10)	0.90(3)	0.34(19)	0.75(9)	0.89(5)	0.41(20)

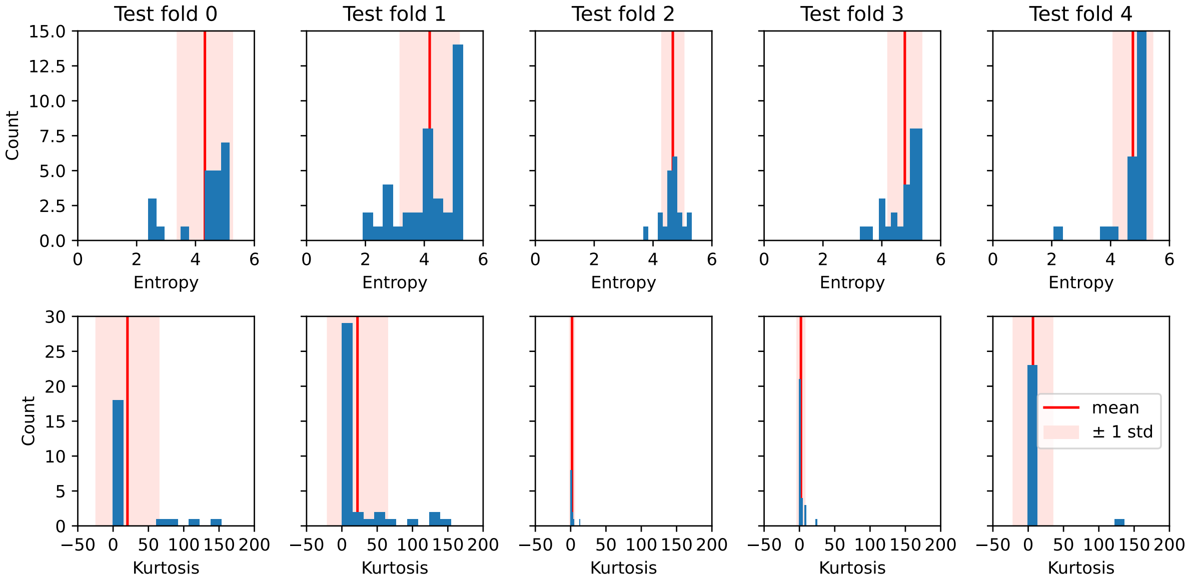


Figure 4.14: Quality of data per test split. Top row: entropy of the tail of the power spectrum. Bottom row: kurtosis of the tail of the power spectrum. The mean (solid) and standard deviation (red region) are shown.

is weighted with a substantial attention weight, resulting in rather dark images.

Location embeddings

A t-SNE projection of the location embeddings is shown in Figure 4.16. Texts seem grouped (*e.g.*, "ventricle", "posterior cranial fossa", "cereb-", "brainstem", "lobe"). The groupings are visualized by fitting a Gaussian mixture model with six components to the t-SNE projections.

4.4.6 Usability

The prediction model can be used intraoperatively to predict tumor type and amount in a biopsy. The biopsy can be placed on the scanner as described in [61] and optionally the location of the tumor can be given in natural language. The model outputs a prediction in seconds.

[61]: Spies (2023), *Validation of higher harmonic generation microscopy for the diagnosis of various pediatric tumors*

To integrate the model with the target system, the raw data needs to be converted to images of 0.2 mpp for the model to accept it. A user interface should be designed with an optional user input for clinical context. All tumors the model has been trained on with their probabilities should be displayed as output. The min-max-normalized attention map should be displayed along the prediction, optionally with a variable threshold. The pipeline should be automated. In particular, the separation between feature vector creation and using them with MIL model should be closed.

Data polluted with blood or a malfunctioning imaging system are not detected by the model. The user should proceed with caution if any of such artifacts appear.

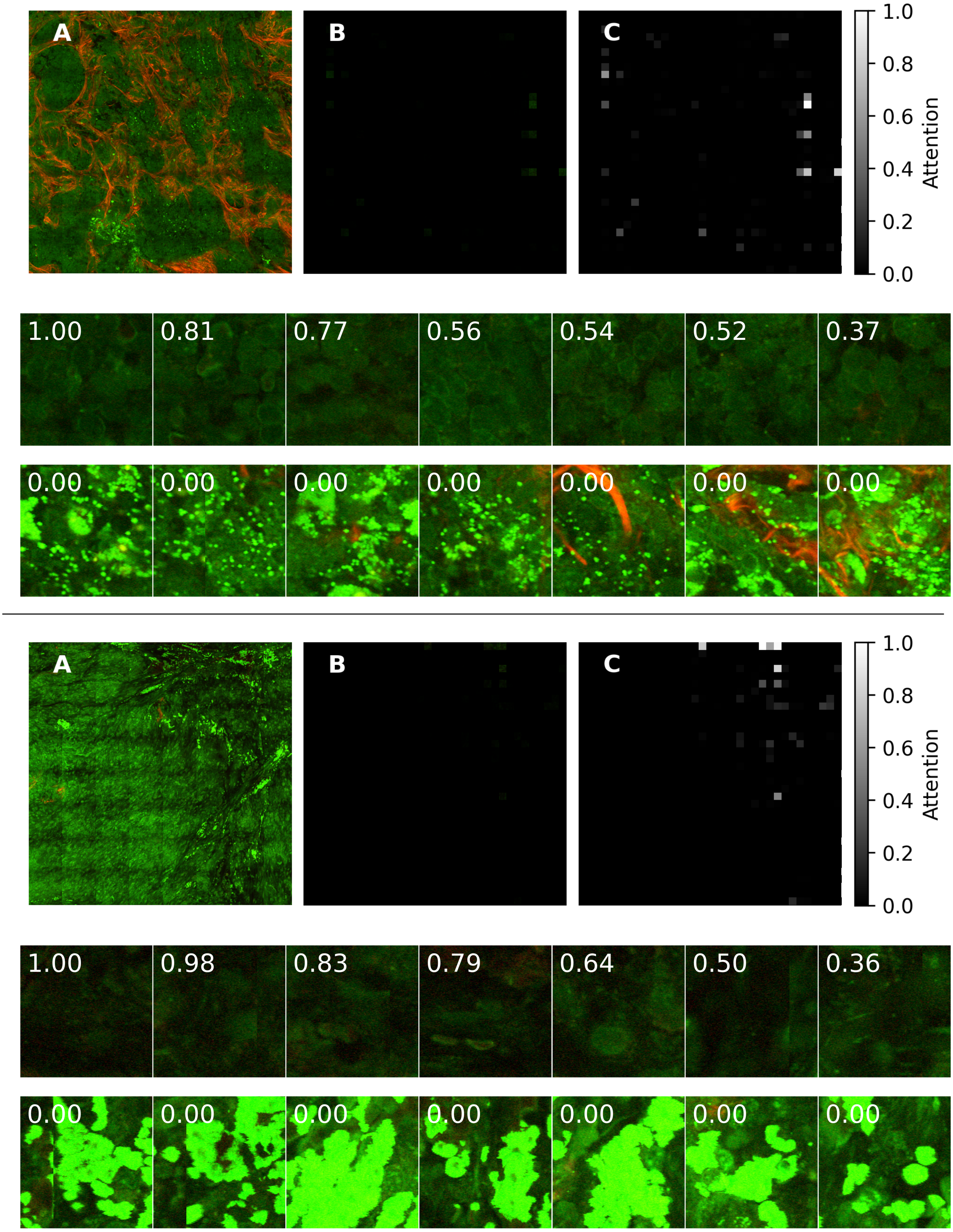


Figure 4.15: Attention weighted images. Tiles are multiplied with min-max-normalized attention weights. Top: medulloblastoma (prediction score = 0.76). Bottom: pilocytic astrocytoma (prediction score = 0.60). First rows show the original image (A), the attention weighted image (B) and the local attentions (linear grayscale) (C). Second and third rows show the tiles with the highest and lowest attention weights, respectively. The corresponding normalized attention weight is printed.

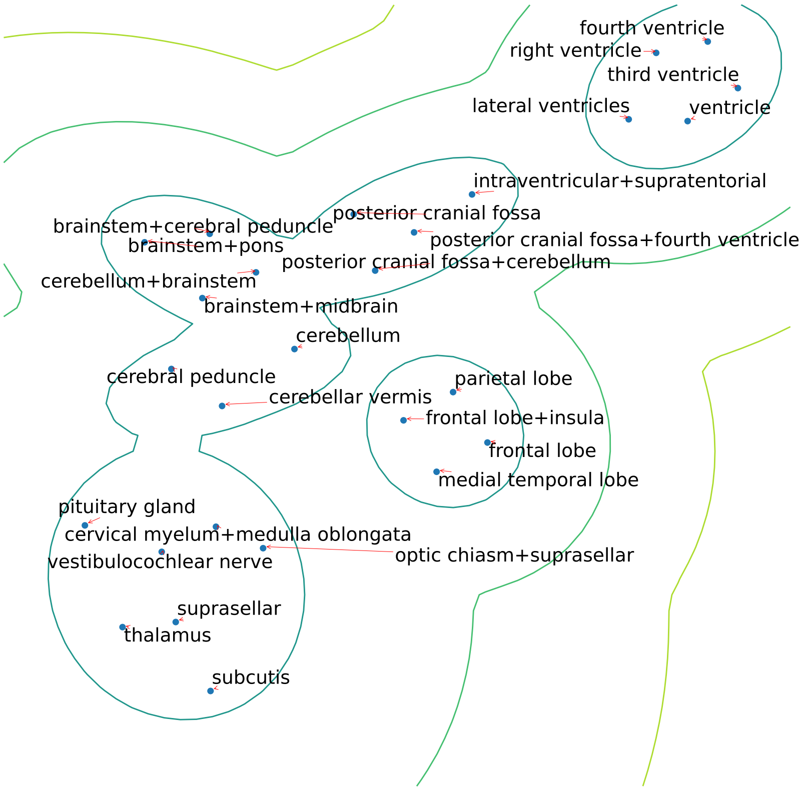


Figure 4.16: T-SNE projections (perplexity = 8, 10^4 iterations) of location embeddings. Texts are connected to points with arrows. A contour plot of a six-component Gaussian mixture model to show groups of text embeddings.

4.5 Discussion

4.5.1 EntropyMasker outperforms (Improved) FESI on HHG data

The masks generated by EntropyMasker show a significant improvement of 44 % compared to FESI and Improved FESI. A possible reason for FESI to perform worse is the tissue oftentimes touching the image boundary, hindering flood fill to determine the complete tissue boundary. Calculating local entropy is not affected by this, as local entropy can still be calculated from pixels at the edge.

However, EntropyMasker also selects fluid-air interfaces. These interfaces can sometimes occur far away from the actual tissue. Still, the probability of including more useful information is higher when selecting more pixels than necessary.

4.5.2 SimCLR clusters features that represent similar structures

The nearest neighbors in image space calculated from the feature representations of the tiles show that SimCLR clusters features that are similar. This is important, as it is assumed that disease features in feature space are similar in the same way they are similar in image space.

Ideally, SimCLR maps tiles to features in as many clusters as there are target classes, such that the classes can be easily separated. In the t-SNE projections (Figure 4.11), the classes seem reasonably well separated.

However, feature projection clusters belonging to the same diagnosis seem to correlate with case number or image number, showing that this self-supervised approach is not able to generalize well. This can likely be improved by using larger feature extraction backbones or more sophisticated self-supervised training scheme like SwAV [85]. Also, the influence of the transformation distribution is not investigated. Possibly, some transformations obfuscate important tumor information while others generate no contrast.

[85]: Caron et al. (2020), *Unsupervised Learning of Visual Features by Contrasting Cluster Assignments*

4.5.3 The dataset is too small to distinguish model performance

The standard errors on the test AUPRG are 62 % (SE 22 %), which is most probably the result of a small dataset. Ref. [60] has shown a 130 % increase in AUC when including five times as much data. With a larger number of samples, there is a higher probability that features of training and testing data are similar, which ought to improve performance.

[60]: Schirris et al. (2022), *DeepSMILE: Contrastive self-supervised pre-training benefits MSI and HRD classification directly from H&E whole-slide images in colorectal and breast cancer*

Future studies could investigate if multiple imaging modalities can be combined via transfer learning [**Zhuang2019**]. Comparable with synthesizing CT images from MRI with deep learning [86], HE images might be synthesized from HHG images. With transfer learning, HE pretrained models can be further fine-tuned on HE transformed HHG data.

Zhuang2019

[86]: Li et al. (2021), *Synthesizing CT images from MR images with Deep Learning: Model generalization for different datasets through transfer learning*

4.5.4 Domain-specific and ImageNet pretrained feature extractors have comparable performance

No evidence is found that domain-specific feature extractors perform better than ImageNet pretrained feature extractors. Model performance on the test and validation set is comparable as the AUC, AUPR, and AUPRG are not significantly different. It is expensive (in terms of time, price and environment) to train a domain-specific feature extractor. Therefore, it is striking that a SimCLR pretrained backbone does not outperform the ImageNet pretrained backbone.

4.5.5 High and low attention seems to be given to appropriate features

High attention medulloblastoma tiles seem to show high cellularity, as expected. High attention pilocytic astrocytoma tiles seem to hardly contain any piloid cells, except maybe in the 0.36 tile. The 0.79 attention tile might contain Rosenthal fibers. All low attention tiles contain very bright structures in THG signal, supposedly mostly produced by blood. Practically discarding these tiles seems reasonable.

4.5.6 Limitations

Models of fold 1 were overfit

The validation loss diverges from the training loss for all models concerning fold 1. This indicates overfitting of the model on the training data of fold 1. One reason for this might be the slightly poorer data quality corresponding to the test set of fold 1. The lower mean entropy and higher mean kurtosis indicate images have less information and the set has more outliers compared to test sets of other folds. The correlation coefficients between entropy, kurtosis, and AUPRG further support this: an increase in entropy means an increase in quality, and a decrease in kurtosis means an increase in quality. Future studies should vary the number of training examples, discarding images of low quality determined by entropy or kurtosis, as described in 3.3.1.

The generalizability issue can also have its origin in images showing the disease with varying features across splits. That is, if some images show *e.g.* cystic areas indicating pilocytic astrocytoma and none of them are in the training set, then it is harder for the model to find those features in the test set. Having a higher sample size reduces the chance of separating disease features into splits, which might improve generalizability.

Tile size was not varied

The HHG microscope can image tissue with a resolution of 0.2 mpp. The tiles that are presented to the model are 44.8 µm×44.8 µm. Medulloblastomas are characterized by the absence of increased cell size (max. ~32 µm [87]), among others. This is smaller than the tile size. Pilocytic astrocytomas develop from astrocytes and their processes are about 97.9 µm [88], which is larger than the tile size. It might be beneficial for the model to work with tile sizes larger than key disease features, otherwise the model may have more difficulty recognizing specific disease patterns. In a future study, the effect of using tiles that are about the size of disease features should be studied.

[87]: Orr (2020), *Pathology, diagnostics, and classification of Medulloblastoma*

[88]: Vasile et al. (2017), *Human astrocytes: Structure and functions in the healthy brain*

Different attention scaling

The attention weighted images seems to highlight just a few of many tiles that a pathologist would use to diagnose. To highlight more tiles, the attention weighted images could be scaled logistically.

A future experiment could also train a model to predict an attention threshold to optimally mask interesting regions annotated by a pathologist. This can be used to reject all areas that are uninteresting, which can be used to accelerate diagnosis by a pathologist.

Clinical context embedding might benefit from fine-tuning

No evidence is found that embedding clinical contexts with an NLP improves performance. Although the AUPRG for both the validation and test set are larger for CCMIL than VarMIL, the difference is insignificant. The language model is frozen during training, meaning that every iteration, the loss does not influence the generation of the text embedding. The weights after the MIL aggregate are learnable. These weights are accountable for the new performance. The text embeddings may be more optimally clustered for this classification task. To possibly achieve this, the NLP can be fine-tuned while training the classifier.

Classifier training batch size of one

The classifier is trained with a batch size of one. However, [60] managed to train VarMIL with a batch size larger than one using padding of the images to overcome the differences in image size. A larger batch size supposedly improves training speed and convergence.

[60]: Schirris et al. (2022), *DeepSMILE: Contrastive self-supervised pre-training benefits MSI and HRD classification directly from H&E whole-slide images in colorectal and breast cancer*

High and low attention tiles were not reviewed by a pathologist

CCMIL outputs attention weights corresponding to input tiles. Tiles with the highest attention should show tumor features while low attention tiles should contain normal tissue or features not important for tumor classification. No pathologist has verified this. Future studies should investigate the association between tumor features found by multiple pathologists and the tile attention.

4.5.7 Implications

Attention weighted images as model explanation

There is a growing need for AI models in the clinical to be explainable. Explainability increases confidence in such models. Although the attention weighted images have few highlighted tiles, they can still be used to assess the prediction quality.

Attention weighted images might be used for visual guidance

Intraoperative tumor type diagnosis can change how surgery is performed, *e.g.*, some tumors types require complete removal but have not been recognized as such before surgery. As the intraoperative HHG images can be of the order of centimeters and the tumor features of the order of micrometers, it is time-expensive for a pathologist to look at the complete image. Highlighting certain areas for the pathologist to look at first might decrease the time spent on intraoperative diagnosis.

Prediction may be consulted as validation

When working with time constraints, such as in an intraoperative setting, human mistakes may occur more frequently. Moreover, pathology using HHG microscopy is not well-established yet, so pathologists would need to be trained on HHG images [61], and therefore they make more errors in the beginning of using this modality. If the model is further improved until a desired performance is reached, the prediction may serve as a (non-binding) diagnosis validation.

[61]: Spies (2023), *Validation of higher harmonic generation microscopy for the diagnosis of various pediatric tumors*

Textual records can be included

If the language model is fine-tuned and CCMIL has proven high precision on a larger testing set, more textual records can be included to the language model. Other data that might be important are *e.g.* prescribed medications or any observations by healthcare specialists.

4.6 Conclusion

The goal of this study was to develop a classifier to distinguish pilocytic astrocytoma and medulloblastoma in higher harmonic generation images while also providing resection location as clinical context. An attention-based multi-instance learning classifier pretrained with SimCLR and using BERT for clinical context embedding achieved a mean average precision of 0.89 (SE 0.05) and 0.41 (SE 0.20) AUPRG. More data is needed to test if the proposed model performs better than an ImageNet pre-trained model or a model without clinical context embedding performed. Although the model could benefit from more data and may be fine-tuned on a broader range of tumors, the model may be used intraoperatively to validate medulloblastoma or pilocytic astrocytoma diagnoses or to pre-select interesting regions for diagnosis.

4.7 Supplementary materials

4.7.1 Code

The implementation of SCLICOM can be found at siemdejong/sclicom.

4.7.2 Data

HHG data is not available. The SLICOM model weights can be downloaded from siemdejong/sclicom.

5
General discussion and conclusion

Combining state-of-the-art subcellular resolution microscopy techniques such as higher harmonic generation microscopy with tailor-made artificial intelligence allows for pattern recognition in biological tissue. Pattern recognition in HHG images can be useful for both regression and classification tasks.

Before developing AI models, it is important to gather data with enough reoccurring features, while also having a broad variety. The data should contain as few artifacts as possible. The data should be split in comparable training and test datasets which should be an accurate sample of the underlying distribution. Developing an AI model to accompany HHG microscopy while data is still to be gathered is challenging, because model development requires a substantial training and validation dataset. If few data is available, data augmentation or transfer learning could help artificially increase the variability and reoccurring features.

Model input may include more than only one type, such as text next to images. All data available to a user can be made available to a replacing model, given that the data is still available at inference, does not induce bias and there is no ambition to use less data.

Given the training data resembles inference data and the model is well-trained, AI models are fast and might eventually replace time-consuming and error-prone human work such as measuring stress-strain curves or diagnosing tumors.

Acknowledgments

I would like to express my deepest appreciation to dr. rer. nat. Patrick González and prof. dr. Marloes Groot for their guidance, feedback, and critical questions.

I am also thankful to Mengyao Zhou, and Sylvia Spies who have provided me with higher harmonic generation images of skin tissue and pediatric brain tumors. Also thanks to the Princess Maxima Center for pediatric oncology making data sharing possible and special thanks to prof. dr. Pieter Wesseling for giving clinical insight.

Lastly, I'd like to acknowledge the group of prof. dr. Marloes Groot and the Biophotonics and Medical Imaging section of the VU for inspiring discussions and feedback.

References

[1] Gary S. Collins et al. 'Transparent Reporting of a multivariable prediction model for Individual Prognosis Or Diagnosis (TRIPOD): The TRIPOD Statement'. In: *Annals of Internal Medicine* 162.1 (Jan. 2015), pp. 55–63. DOI: 10.7326/M14-0697. (Visited on 09/19/2022) (cited on page 2).

[2] Karel G.M. Moons et al. 'Transparent reporting of a multivariable prediction model for individual prognosis or diagnosis (TRIPOD): Explanation and elaboration'. In: *Annals of Internal Medicine* 162.1 (2015). DOI: 10.7326/m14-0698 (cited on page 2).

[3] Pauline Heus et al. 'Transparent reporting of multivariable prediction models in Journal and conference abstracts: Tripod for abstracts'. In: *Annals of Internal Medicine* 173.1 (2020), pp. 42–47. DOI: 10.7326/m20-0193 (cited on page 2).

[4] Gary S Collins et al. 'Protocol for development of a reporting guideline (TRIPOD-AI) and risk of bias tool (PROBAST-AI) for Diagnostic and prognostic prediction model studies based on Artificial Intelligence'. In: *BMJ Open* 11.7 (2021). DOI: 10.1136/bmjopen-2020-048008 (cited on page 3).

[5] Gary Collins et al. *TRIPOD-AI*. Sept. 2020. URL: https://osf.io/zyacb/ (cited on page 3).

[6] Robert F. Wolff et al. 'PROBAST: A tool to assess the risk of bias and applicability of Prediction model studies'. In: *Annals of Internal Medicine* 170.1 (2019), p. 51. DOI: 10.7326/m18-1376 (cited on page 3).

[7] Karel G.M. Moons et al. 'PROBAST: A tool to assess risk of bias and applicability of prediction model studies: Explanation and elaboration'. In: *Annals of Internal Medicine* 170.1 (2019). DOI: 10.7326/m18-1377 (cited on page 3).

[8] *The nobel prize in physiology or medicine 1981*. Oct. 1981. URL: https://www.nobelprize.org/prizes/medicine/1981/press-release/ (cited on page 5).

[9] Wulfram Gerstner and Werner M. Kistler. *Spiking Neuron Models: Single Neurons, Populations, Plasticity*. Cambridge University Press, 2002 (cited on page 6).

[10] Kunihiko Fukushima. 'Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position'. In: *Biological Cybernetics* 36.4 (1980), pp. 193–202. DOI: 10.1007/bf00344251 (cited on page 5).

[11] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. 'Learning representations by back-propagating errors'. In: *Nature* 323.6088 (1986), pp. 533–536. DOI: 10.1038/323533a0 (cited on pages 5, 6).

[12] Yann Le Cun et al. 'Handwritten digit recognition with a back-propagation network'. English (US). In: *Advances in Neural Information Processing Systems (NIPS 1989), Denver, CO*. Ed. by David Touretzky. Vol. 2. Morgan Kaufmann, 1990 (cited on page 5).

[13] Diederik P. Kingma and Jimmy Ba. 'Adam: A Method for Stochastic Optimization'. In: *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. Ed. by Yoshua Bengio and Yann LeCun. 2015 (cited on page 6).

[14] Vincent Dumoulin and Francesco Visin. *A guide to convolution arithmetic for deep learning*. 2016. DOI: 10.48550/ARXIV.1603.07285. URL: https://arxiv.org/abs/1603.07285 (cited on page 8).

[15] Xiankai Lu et al. 'Deep Object Tracking With Shrinkage Loss'. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44.5 (May 2022), pp. 2386–2401. DOI: 10.1109/TPAMI.2020.3041332 (cited on page 10).

[16] Nitish Srivastava et al. 'Dropout: A Simple Way to Prevent Neural Networks from Overfitting'. In: *J. Mach. Learn. Res.* 15.1 (2014), pp. 1929–1958 (cited on page 11).

[17] Sergey Ioffe and Christian Szegedy. 'Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift'. In: *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*. ICML'15. Lille, France: JMLR.org, 2015, pp. 448–456 (cited on page 11).

[18] Johan Bjorck, Carla P. Gomes, and Bart Selman. 'Understanding Batch Normalization'. In: *CoRR* abs/1806.02375 (2018) (cited on page 12).

[19] James Bergstra and Yoshua Bengio. 'Random Search for Hyper-Parameter Optimization'. In: *J. Mach. Learn. Res.* 13 (2012), pp. 281–305 (cited on page 12).

[20] James Bergstra et al. 'Algorithms for Hyper-Parameter Optimization'. In: *Proceedings of the 24th International Conference on Neural Information Processing Systems*. NIPS'11. Granada, Spain: Curran Associates Inc., 2011, pp. 2546–2554 (cited on page 13).

[21] Stefan Falkner, Aaron Klein, and Frank Hutter. 'BOHB: Robust and Efficient Hyperparameter Optimization at Scale'. In: *Proceedings of the 35th International Conference on Machine Learning*. Ed. by Jennifer Dy and Andreas Krause. Vol. 80. Proceedings of Machine Learning Research. PMLR, Oct. 2018, pp. 1437–1446 (cited on page 13).

[22] Donald R. Jones. 'A Taxonomy of Global Optimization Methods Based on Response Surfaces'. In: *J. of Global Optimization* 21.4 (2001), pp. 345–383. DOI: 10.1023/A:1012771025575 (cited on page 13).

[23] Kevin Jamieson and Ameet Talwalkar. 'Non-stochastic Best Arm Identification and Hyperparameter Optimization'. In: *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*. Ed. by Arthur Gretton and Christian C. Robert. Vol. 51. Proceedings of Machine Learning Research. Cadiz, Spain: PMLR, Sept. 2016, pp. 240–248 (cited on page 13).

[24] Lisha Li et al. 'Hyperband: A Novel Bandit-Based Approach to Hyperparameter Optimization'. In: *J. Mach. Learn. Res.* 18.1 (2017), pp. 6765–6816 (cited on page 13).

[25] Frank Hutter, Holger Hoos, and Kevin Leyton-Brown. 'An Efficient Approach for Assessing Hyperparameter Importance'. In: *Proceedings of the 31st International Conference on Machine Learning*. Ed. by Eric P. Xing and Tony Jebara. Vol. 32. Proceedings of Machine Learning Research 1. Bejing, China: PMLR, June 2014, pp. 754–762 (cited on page 14).

[26] Max Blokker et al. 'Fast intraoperative histology-based diagnosis of gliomas with third harmonic generation microscopy and deep learning'. In: *Scientific Reports* 12.1 (2022), pp. 11334–11334. DOI: 10.1038/s41598-022-15423-z (cited on pages 14, 15, 25, 51).

[27] Sami Koho et al. 'Image Quality Ranking Method for Microscopy'. In: *Scientific Reports* 6 (July 2016), p. 28962. DOI: 10.1038/srep28962. (Visited on 11/09/2022) (cited on pages 14, 15, 25).

[28] Matthew D. Zeiler and Rob Fergus. 'Visualizing and Understanding Convolutional Networks'. In: *Computer Vision – ECCV 2014*. Ed. by David Fleet et al. Cham: Springer International Publishing, 2014, pp. 818–833 (cited on page 15).

[29] Mengyao Zhou et al. 'Three-dimensional Characterization of Mechanical Properties and Microstructures of Human Dermal Skin' (cited on pages 20, 24, 25, 31).

[30] Pauline D. Verhaegen et al. 'Adaptation of the dermal collagen structure of human skin and scar tissue in response to stretch: An experimental study'. In: *Wound Repair and Regeneration* 20.5 (2012), pp. 658–666. DOI: 10.1111/j.1524-475x.2012.00827.x (cited on page 20).

[31] Alperen Soylu. 'Developing a non-invasive approach to estimate physical parameters of skin tissue from HHG microscopy using convolutional neural networks'. Unpublished. MSc thesis. Vrije Universiteit Amsterdam, 2022 (cited on pages 20, 22, 26, 28, 31).

[32] Gerhard A. Holzapfel. 'Biomechanics of Soft Tissue'. In: *Handbook of Materials Behavior Models*. Ed. by Jean Lemaitre. Burlington: Academic Press, 2001, pp. 1057–1071. DOI: https://doi.org/10.1016/B978-012443341-0/50107-1 (cited on pages 20, 22).

[33] *OriginPro*. Version 2022b. Northampton, MA, USA: OriginLab Corporation, 2022 (cited on page 22).

[34] Yuzhe Yang et al. 'Delving into Deep Imbalanced Regression'. In: *Proceedings of the 38th International Conference on Machine Learning*. Ed. by Marina Meila and Tong Zhang. Vol. 139. Proceedings of Machine Learning Research. PMLR, 18–24 Jul 2021, pp. 11842–11851 (cited on page 23).

[35] L van Haasterecht et al. 'Visualizing dynamic Three-dimensional changes of human reticular dermal collagen under mechanical strain'. In: *Biomedical Physics and Engineering Express* 9.3 (2023), p. 035033. DOI: 10.1088/2057-1976/accc8e (cited on pages 24, 31).

[36] Karel Zuiderveld. 'Contrast Limited Adaptive Histogram Equalization'. In: *Graphics Gems IV*. USA: Academic Press Professional, Inc., 1994, pp. 474–485 (cited on page 24).

[37] Stéfan van der Walt et al. 'scikit-image: image processing in Python'. In: *PeerJ* 2 (June 2014), e453. DOI: 10.7717/peerj.453 (cited on page 24).

[38] Pauli Virtanen et al. 'SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python'. In: *Nature Methods* 17 (2020), pp. 261–272. DOI: 10.1038/s41592-019-0686-2 (cited on pages 24, 26).

[39] Lei Huang et al. 'Normalization Techniques in Training DNNs: Methodology, Analysis and Application'. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2023), pp. 1–20. DOI: 10.1109/TPAMI.2023.3250241 (cited on page 24).

[40] TorchVision maintainers and contributors. *TorchVision: PyTorch's Computer Vision library*. https://github.com/pytorch/vision. 2016 (cited on page 26).

[41] F. Pedregosa et al. 'Scikit-learn: Machine Learning in Python'. In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830 (cited on page 26).

[42] Liang Liang, Minliang Liu, and Wei Sun. 'A deep learning approach to estimate chemically-treated collagenous tissue nonlinear anisotropic stress-strain responses from microscopy images'. In: *Acta Biomaterialia* 63 (2017), pp. 227–235. DOI: https://doi.org/10.1016/j.actbio.2017.09.025 (cited on page 28).

[43] Kaiming He et al. 'Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification'. In: *2015 IEEE International Conference on Computer Vision (ICCV)*. 2015, pp. 1026–1034. DOI: 10.1109/ICCV.2015.123 (cited on page 29).

[44] Takuya Akiba et al. 'Optuna: A Next-Generation Hyperparameter Optimization Framework'. In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. KDD '19. Anchorage, AK, USA: Association for Computing Machinery, 2019, pp. 2623–2631. DOI: 10.1145/3292500.3330701 (cited on pages 29, 64).

[45] Gao Huang et al. 'Snapshot Ensembles: Train 1, Get M for Free'. In: *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017 (cited on page 29).

[46] Adam Paszke et al. 'PyTorch: An Imperative Style, High-Performance Deep Learning Library'. In: *Advances in Neural Information Processing Systems 32*. Ed. by H. Wallach et al. Curran Associates, Inc., 2019, pp. 8024–8035 (cited on page 30).

[47] Narine Kokhlikyan et al. *Captum: A unified and generic model interpretability library for PyTorch*. 2020 (cited on page 30).

[48] Johannes Schindelin et al. 'Fiji: An open-source platform for biological-image analysis'. In: *Nature Methods* 9.7 (2012), pp. 676–682. DOI: 10.1038/nmeth.2019 (cited on page 31).

[49] Katarzyna Lipa et al. 'Does smoking affect your skin?' In: *Advances in Dermatology and Allergology* 38.3 (2021), pp. 371–376. DOI: 10.5114/ada.2021.103000 (cited on page 46).

[50] Holly N. Wilkinson and Matthew J. Hardman. 'Wound healing: cellular mechanisms and pathological outcomes'. In: *Open Biology* 10.9 (2020), p. 200223. DOI: 10.1098/rsob.200223 (cited on page 46).

[51] Alexander Krull, Tim-Oliver Buchholz, and Florian Jug. 'Noise2Void - Learning Denoising From Single Noisy Images'. In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019, pp. 2124–2132. DOI: 10.1109/CVPR.2019.00223 (cited on page 47).

[52] Eva Höck et al. 'N2V2 - Fixing Noise2Void Checkerboard Artifacts With Modified Sampling Strategies And Tweaked Network Architecture'. In: *Computer Vision - ECCV 2022 Workshops: Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part IV*. Tel Aviv, Israel: Springer-Verlag, 2023, pp. 503–518. DOI: 10.1007/978-3-031-25069-9_33 (cited on page 47).

[53] N. V. Kuzmin et al. 'Third harmonic generation imaging for fast, label-free pathology of human brain tumors'. In: *Biomedical Optics Express* 7.5 (2016), pp. 1889–1889. DOI: 10.1364/BOE.7.001889 (cited on page 48).

[54] Jonathan M. Kocarnik et al. 'Cancer incidence, mortality, years of life lost, years lived with disability, and disability-adjusted life years for 29 cancer groups from 2010 to 2019'. In: *JAMA Oncology* 8.3 (2022), p. 420. DOI: 10.1001/jamaoncol.2021.6987 (cited on page 51).

[55] Maral Adel Fahmideh and Michael E. Scheurer. 'Pediatric brain tumors: Descriptive epidemiology, risk factors, and future directions'. In: *Cancer Epidemiology, Biomarkers and Prevention* 30.5 (2021), pp. 813–821. DOI: 10.1158/1055-9965.epi-20-1443 (cited on page 51).

[56] Melissa R. George et al. 'Will I need to move to get my first job?: Geographic relocation and other trends in the pathology job market'. In: *Archives of Pathology and Laboratory Medicine* 144.4 (2019), pp. 427–434. DOI: 10.5858/arpa.2019-0150-cp (cited on page 51).

[57] Anil V. Parwani. 'Next Generation Diagnostic Pathology: Use of digital pathology and artificial intelligence tools to augment a pathological diagnosis'. In: *Diagnostic Pathology* 14.1 (2019). DOI: 10.1186/s13000-019-0921-2 (cited on page 51).

[58] SyedAhmed Taqi et al. 'A review of artifacts in histopathology'. In: *Journal of Oral and Maxillofacial Pathology* 22.2 (2018), p. 279. DOI: 10.4103/jomfp.jomfp_125_15 (cited on page 51).

[59] Geert Litjens et al. 'A survey on deep learning in medical image analysis'. In: *Medical Image Analysis* 42 (2017), pp. 60–88. DOI: https://doi.org/10.1016/j.media.2017.07.005 (cited on page 51).

[60] Yoni Schirris et al. 'DeepSMILE: Contrastive self-supervised pre-training benefits MSI and HRD classification directly from H&E whole-slide images in colorectal and breast cancer'. In: *Medical Image Analysis* 79 (2022), pp. 102464–102464. DOI: 10.1016/j.media.2022.102464 (cited on pages 51, 55, 74, 76).

[61] Sylvia Spies. 'Validation of higher harmonic generation microscopy for the diagnosis of various pediatric tumors'. Unpublished. MSc thesis. University of Amsterdam and VU Amsterdam, 2023 (cited on pages 52, 60, 71, 76).

[62] Ting Chen et al. 'A Simple Framework for Contrastive Learning of Visual Representations'. In: *Proceedings of the 37th International Conference on Machine Learning*. Ed. by Hal Daumé III and Aarti Singh. Vol. 119. Proceedings of Machine Learning Research. PMLR, 13–18 Jul 2020, pp. 1597–1607 (cited on pages 52, 53).

[63] Maximilian Ilse, Jakub Tomczak, and Max Welling. 'Attention-based Deep Multiple Instance Learning'. In: *Proceedings of the 35th International Conference on Machine Learning*. Ed. by Jennifer Dy and Andreas Krause. Vol. 80. Proceedings of Machine Learning Research. PMLR, Oct. 2018, pp. 2127–2136 (cited on page 54).

[64] Nathan E. Millard and Kevin C. De Braganca. 'Medulloblastoma'. In: *Journal of Child Neurology* 31.12 (2016), pp. 1341–1353. DOI: 10.1177/0883073815600866 (cited on page 56).

[65] Faiza Khan Khattak et al. 'A survey of word embeddings for clinical text'. en. In: *Journal of Biomedical Informatics*. Articles initially published in Journal of Biomedical Informatics: X 1-4, 2019 100 (Jan. 2019), p. 100057. DOI: 10.1016/j.yjbinx.2019.100057. (Visited on 04/05/2023) (cited on page 56).

[66] Ashish Vaswani et al. 'Attention is All you Need'. In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon et al. Vol. 30. Curran Associates, Inc., 2017 (cited on page 56).

[67] Jacob Devlin et al. 'BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding'. In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, June 2019, pp. 4171–4186. DOI: 10.18653/v1/N19-1423 (cited on page 56).

[68] Matthew E. Peters et al. 'Deep Contextualized Word Representations'. In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. New Orleans, Louisiana: Association for Computational Linguistics, June 2018, pp. 2227–2237. DOI: 10.18653/v1/N18-1202 (cited on page 56).

[69] Takaya Saito and Marc Rehmsmeier. 'The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets'. In: *PLOS ONE* 10.3 (Mar. 2015). DOI: 10.1371/journal.pone.0118432 (cited on page 58).

[70] Peter Flach and Meelis Kull. 'Precision-Recall-Gain Curves: PR Analysis Done Right'. In: *Advances in Neural Information Processing Systems*. Ed. by C. Cortes et al. Vol. 28. Curran Associates, Inc., 2015 (cited on pages 58, 59, 64).

[71] Daniel Bug, Friedrich Feuerhake, and Dorit Merhof. 'Foreground extraction for histopathological whole slide imaging'. In: *Informatik aktuell* (2015), pp. 419–424. DOI: 10.1007/978-3-662-46224-9_72 (cited on page 60).

[72] Abtin Riasatian et al. 'A Comparative Study of U-Net Topologies for Background Removal in Histopathology Images'. In: *2020 International Joint Conference on Neural Networks (IJCNN)*. 2020, pp. 1–8. DOI: 10.1109/IJCNN48605.2020.9207018 (cited on page 60).

[73] Yipei Song et al. 'An automatic entropy method to efficiently mask histology whole-slide images'. In: *Scientific Reports* 13.1 (2023). DOI: 10.1038/s41598-023-29638-1 (cited on page 60).

[74] Siem de Jong, Netherlands Cancer Institute, and contributors. *Deep Learning Utilities for Pathology, branch entropy_masker*. 2023 (cited on page 61).

[75] Netherlands Cancer Institute and contributors. *Deep Learning Utilities for Pathology*. Mar. 2023 (cited on page 61).

[76] Peter Bankhead et al. 'QuPath: Open source software for digital pathology image analysis'. In: *Scientific Reports* 7.1 (2017). DOI: 10.1038/s41598-017-17204-5 (cited on page 61).

[77] Ningning Ma et al. 'ShuffleNet V2: Practical Guidelines for Efficient CNN Architecture Design'. In: *Computer Vision – ECCV 2018*. Ed. by Vittorio Ferrari et al. Cham: Springer International Publishing, 2018, pp. 122–138 (cited on page 62).

[78] imgclsmob contributors. *imgclsmob*. https://github.com/osmr/imgclsmob. 2023 (cited on page 62).

[79] Omid Rohanian et al. *Lightweight Transformers for Clinical Natural Language Processing*. 2023. DOI: 10.48550/ARXIV.2302.04725. URL: https://arxiv.org/abs/2302.04725 (cited on page 62).

[80] Thomas Wolf et al. 'Transformers: State-of-the-Art Natural Language Processing'. In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Online: Association for Computational Linguistics, Oct. 2020, pp. 38–45. DOI: 10.18653/v1/2020.emnlp-demos.6 (cited on page 62).

[81] Alistair E.W. Johnson et al. 'Mimic-III, a freely accessible Critical Care Database'. In: *Scientific Data* 3.1 (2016). DOI: 10.1038/sdata.2016.35 (cited on page 62).

[82] William Falcon and The PyTorch Lightning team. *PyTorch Lightning*. Version 1.4. Mar. 2019. DOI: 10.5281/zenodo.3828935 (cited on page 64).

[83] Richard Liaw et al. 'Tune: A Research Platform for Distributed Model Selection and Training'. In: *CoRR* abs/1807.05118 (2018) (cited on page 64).

[84] Nicki Skafte Detlefsen et al. *TorchMetrics - Measuring Reproducibility in PyTorch*. Feb. 2022. DOI: 10.21105/joss.04101 (cited on page 64).

[85] Mathilde Caron et al. 'Unsupervised Learning of Visual Features by Contrasting Cluster Assignments'. In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle et al. Vol. 33. Curran Associates, Inc., 2020, pp. 9912–9924 (cited on page 74).

[86] Wen Li et al. 'Synthesizing CT images from MR images with Deep Learning: Model generalization for different datasets through transfer learning'. In: *Biomedical Physics and Engineering Express* 7.2 (2021), p. 025020. DOI: 10.1088/2057-1976/abe3a7 (cited on page 74).

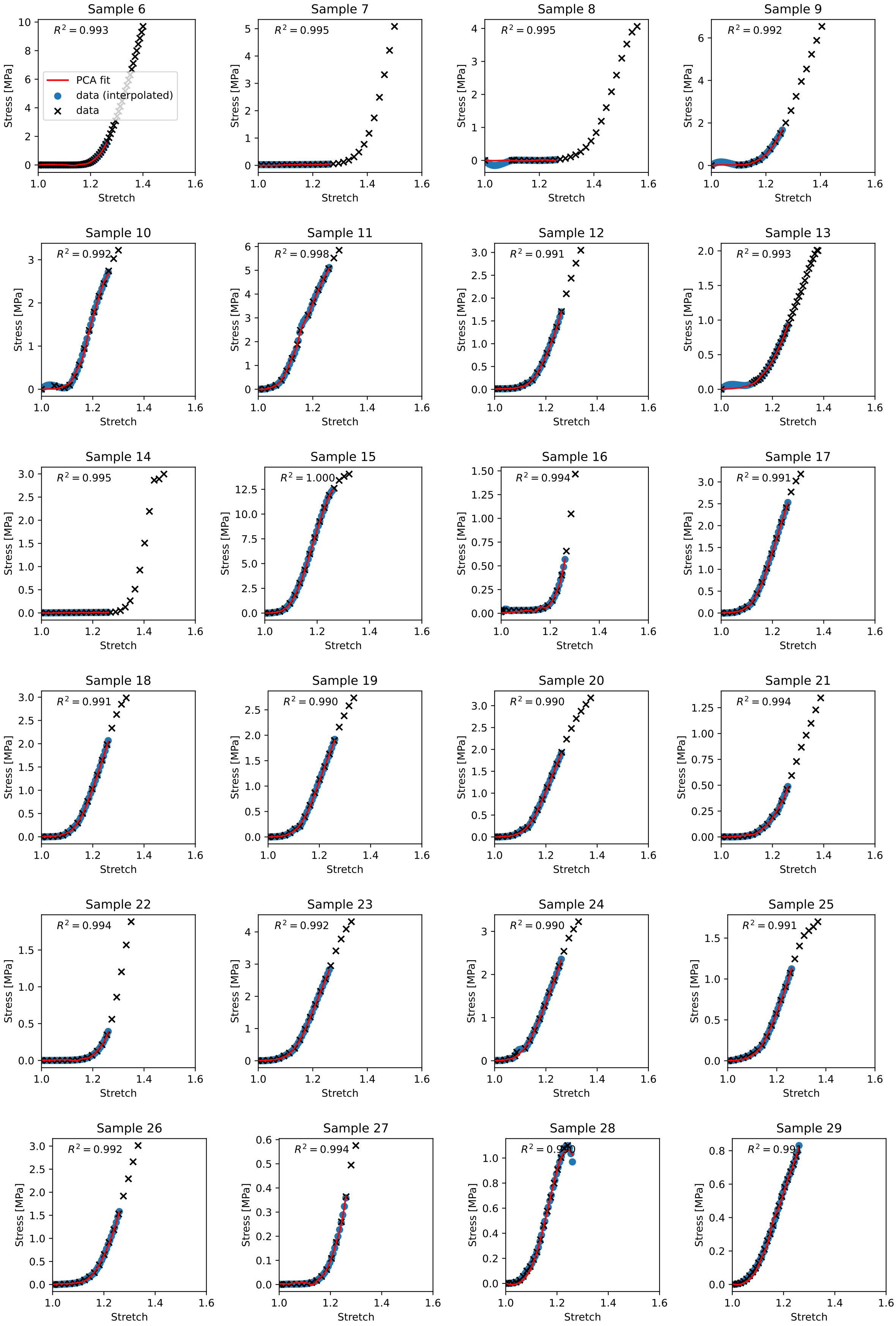
[87] Brent A. Orr. 'Pathology, diagnostics, and classification of Medulloblastoma'. In: *Brain Pathology* 30.3 (2020), pp. 664–678. DOI: 10.1111/bpa.12837 (cited on page 75).

[88] Flora Vasile, Elena Dossi, and Nathalie Rouach. 'Human astrocytes: Structure and functions in the healthy brain'. In: *Brain Structure and Function* 222.5 (2017), pp. 2017–2029. DOI: 10.1007/s00429-017-1383-5 (cited on page 75).

A
Skinstression

A.1 Fits to stress-strain curves

PCA and logistic curve fits to the skin stress-strain curves are shown in Figures A.1 and A.2, respectively.



Continued on next page.

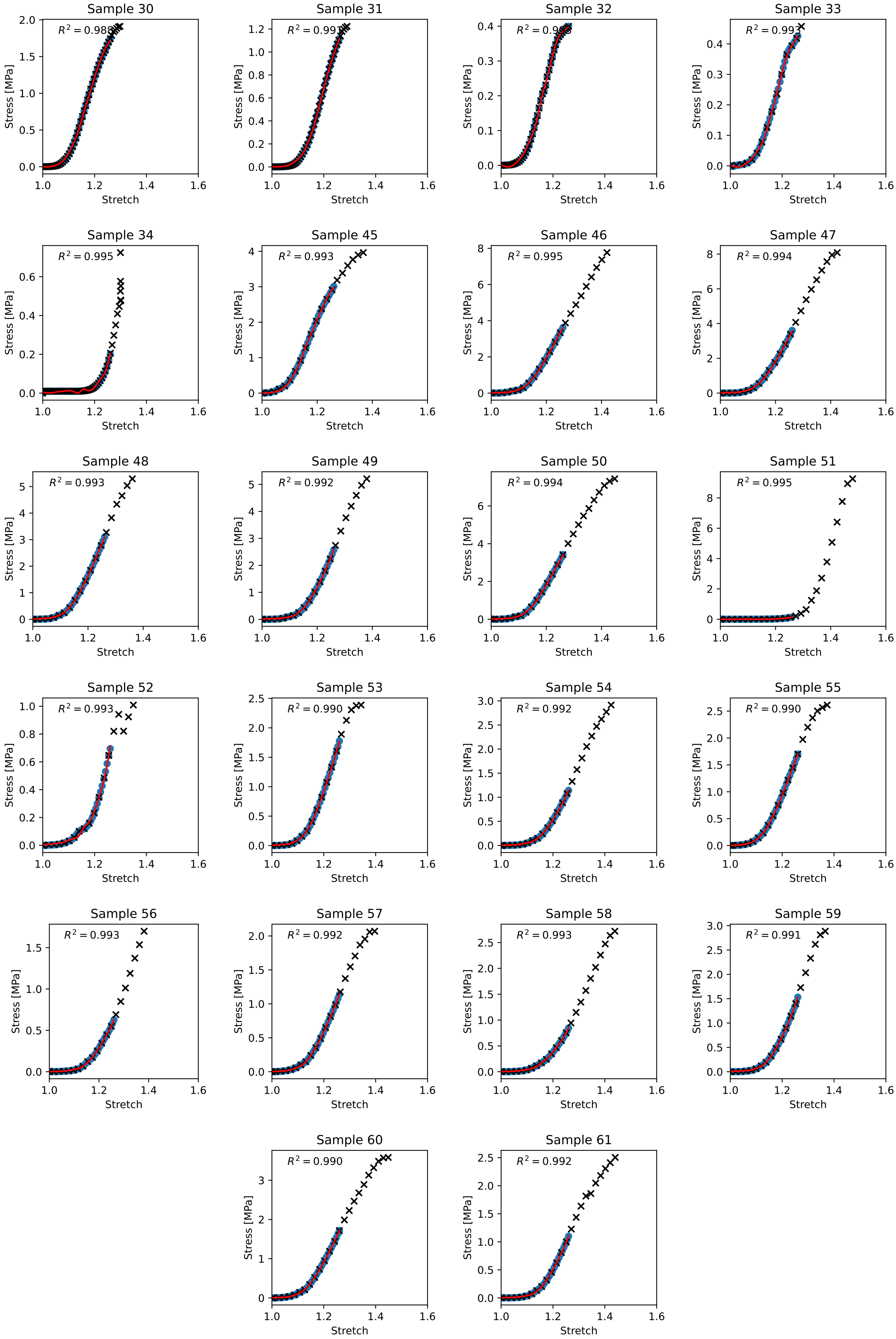
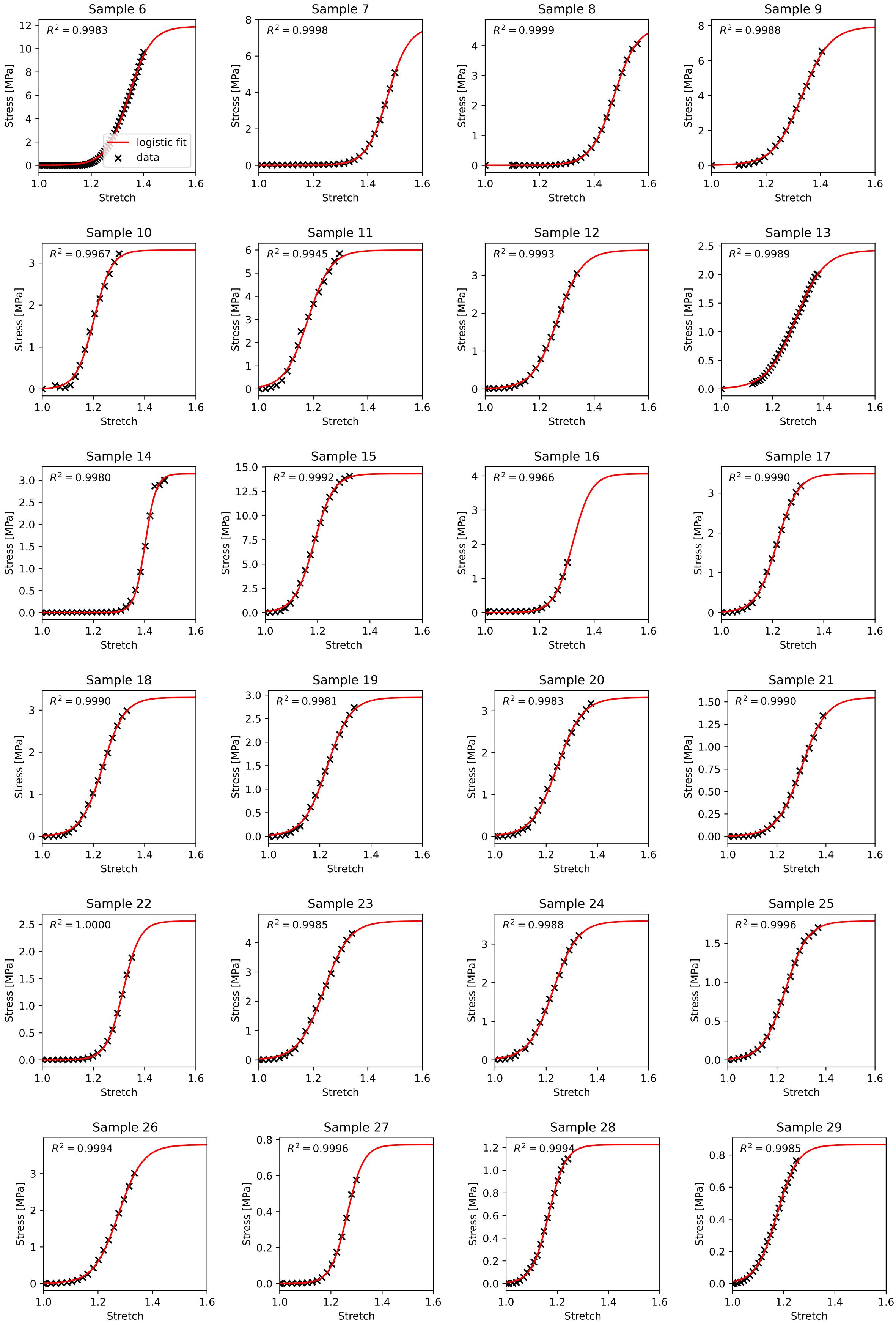


Figure A.1: PCA fits for every truncated and interpolated strain-stress curve. The interpolated measurements (blue) are estimated by the PCA curve (red) along with their R^2. PCA is done on all available thigh data. Note that the vertical axes are not equal.



Continued on next page.

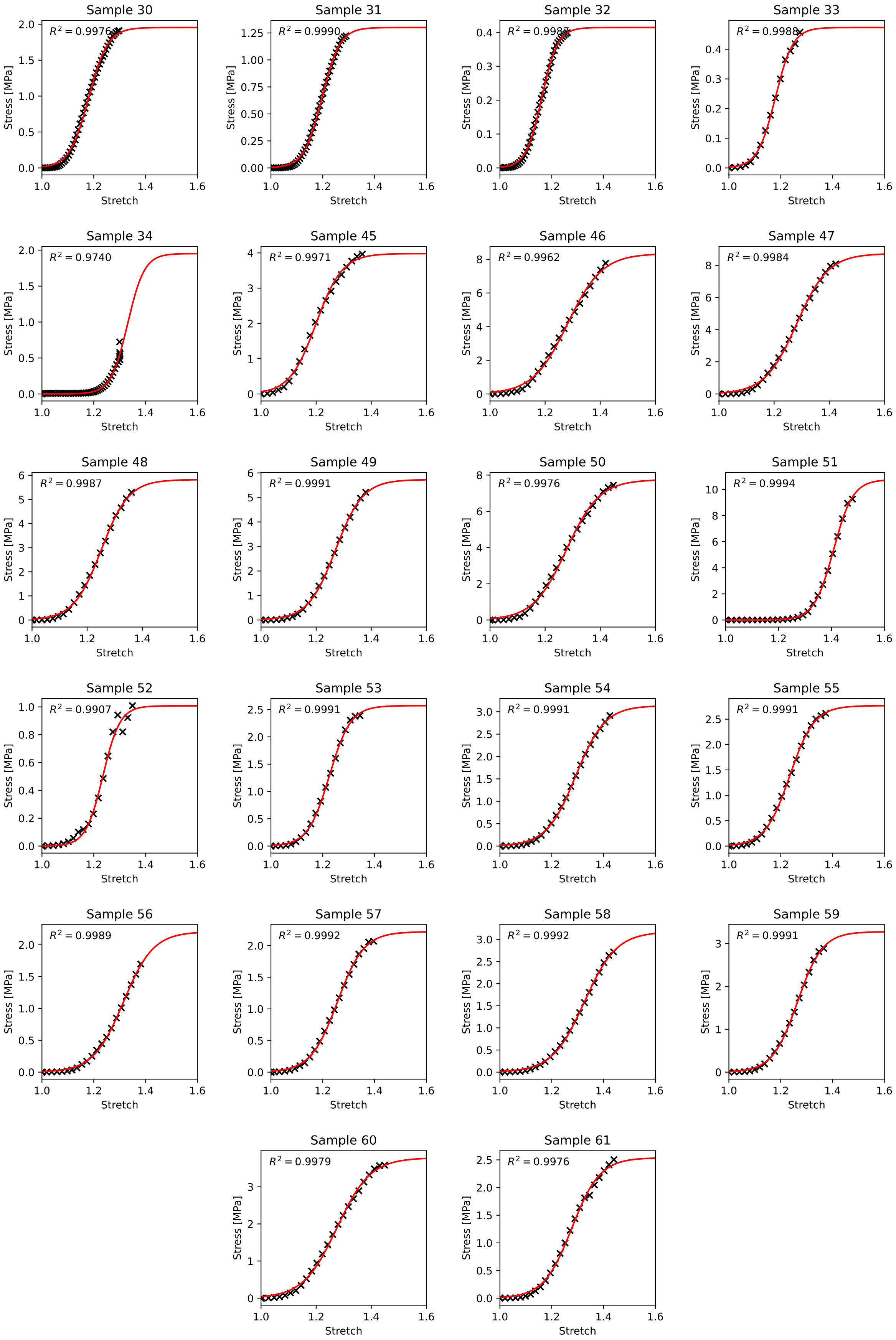


Figure A.2: Logistic fits (red) and their R^2 for every strain-stress curve (black). Note that the vertical axes are not equal.

A.2 Training prediction

Results of a random training batch are shown in Figure A.3.

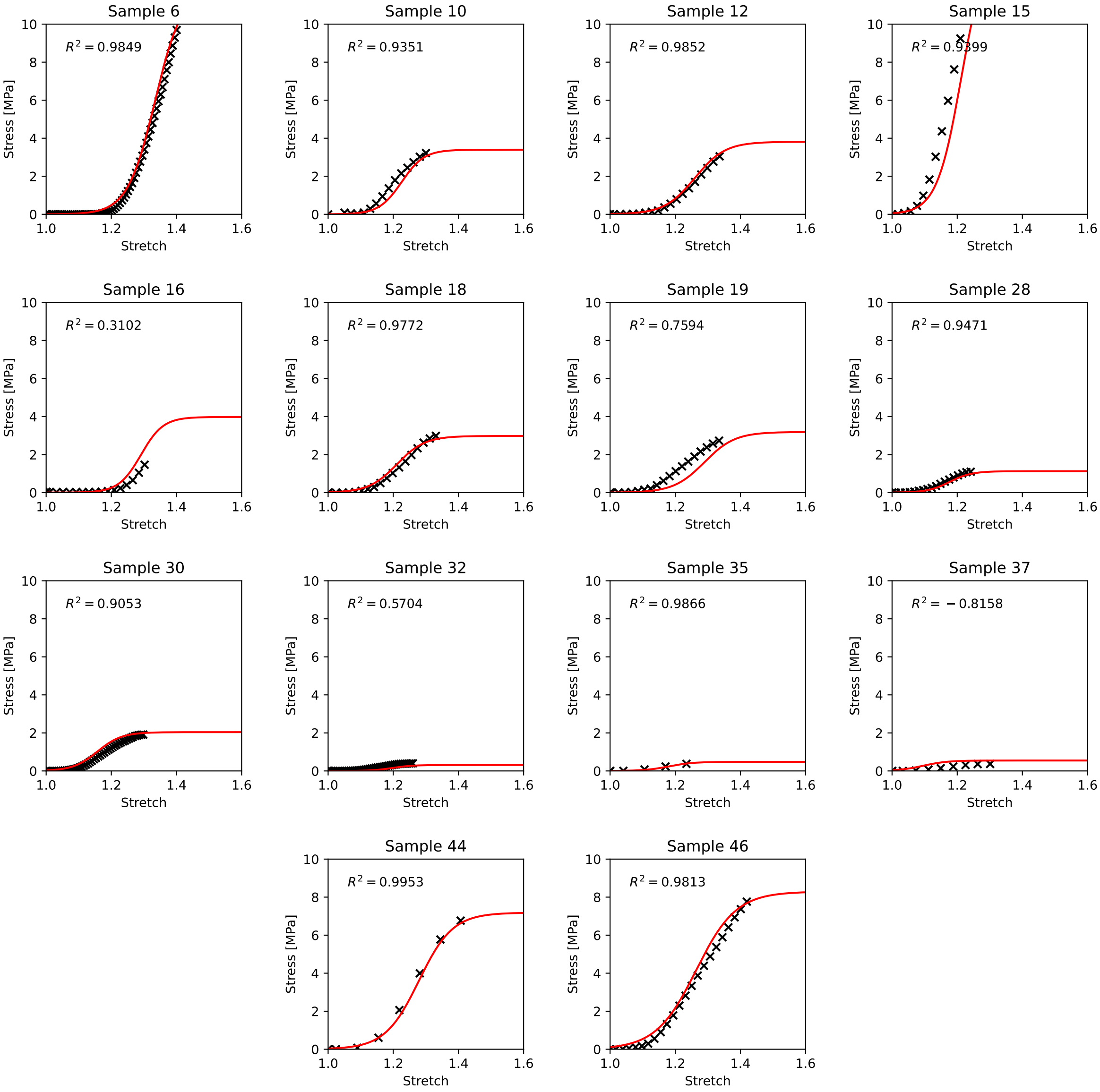


Figure A.3: Results of a random training batch. Raw mechanical measurement without error bars (×) are shown together with the AI fit (red). Performance is quantified by R^2.

A.3 Configuration spaces

The configuration search space for Skinstression is summarized in Table A.1.

Table A.1: Skinstression configuration search space.

parameter	type	min	max	step	log
weight decay	float	10^{-5}	10^{-4}	-	✓
learning rate	float	10^{-6}	10^{-2}	-	✓
T_0	integer	100	300	1	✗
T_{mult}	integer	1	5	1	✗
n_{nodes}	integer	64	128	64	✗
batch size	integer	8	64	8	✗

A.4 Software diagrams

Communicating code can be done using the C4 model. This model is an industry standard to visually communicate software architectures. A map of the architecture can be made on four levels. A higher level zooms in on the previous level. The first level of the C4 model sets the context of the software. The second level shows the high-level technical building blocks of the software.

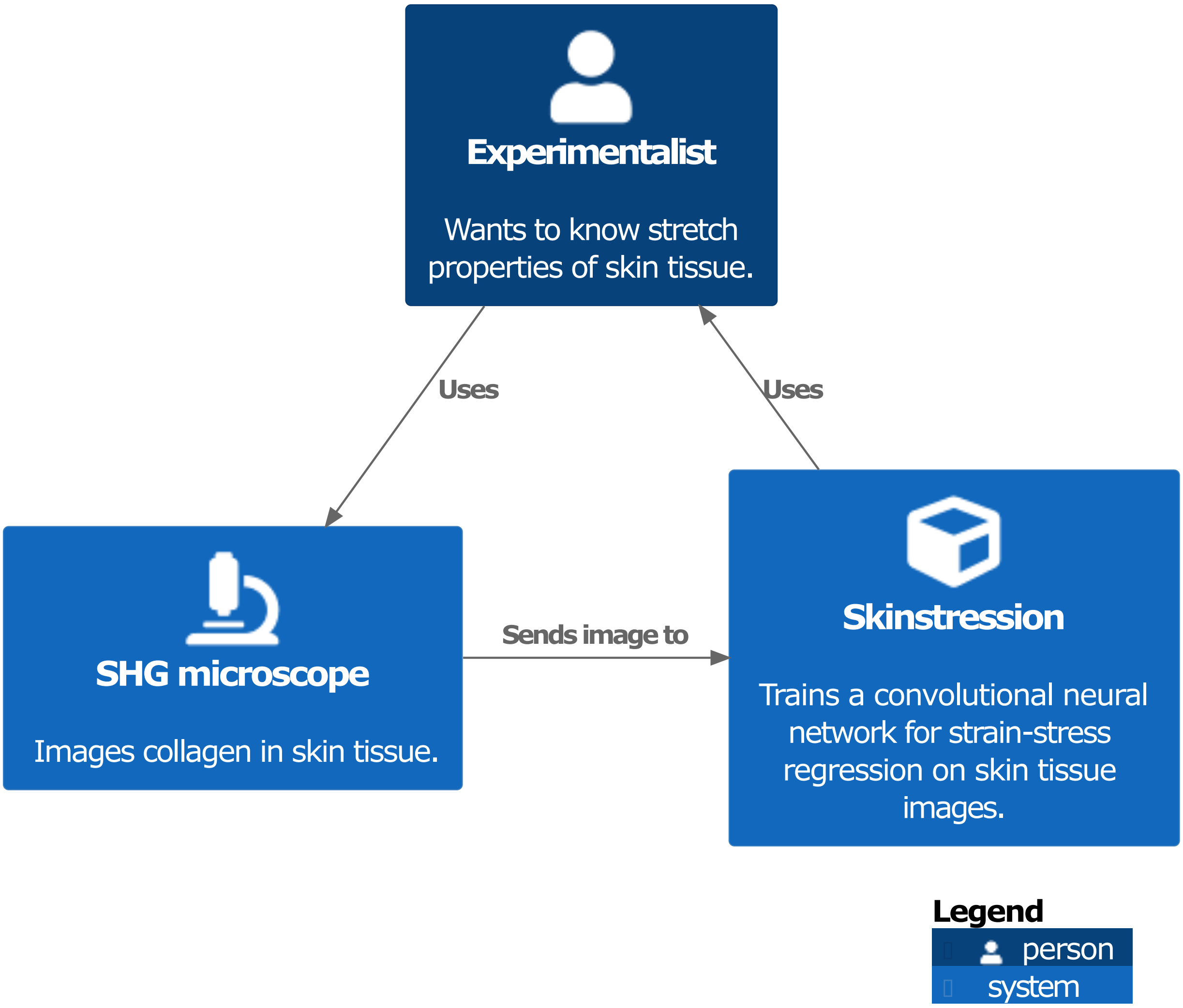

Experimentalist
Wants to know stretch properties of skin tissue.
Uses
Uses
SHG microscope
Images collagen in skin tissue.
Sends image to
Skinstression
Trains a convolutional neural network for strain-stress regression on skin tissue images.
Legend
person
system

Figure A.4: System context diagram of Skinstression. An experimentalist images collagen in skin tissue using an SHG microscope. The microscope output serves as input to Skinstression which trains a convolutional neural network to find the strain-stress curve of the imaged tissue. The trained model can serve as a substitution to the SHG microscope, or provide new insights in why tissue has particular stretch properties.

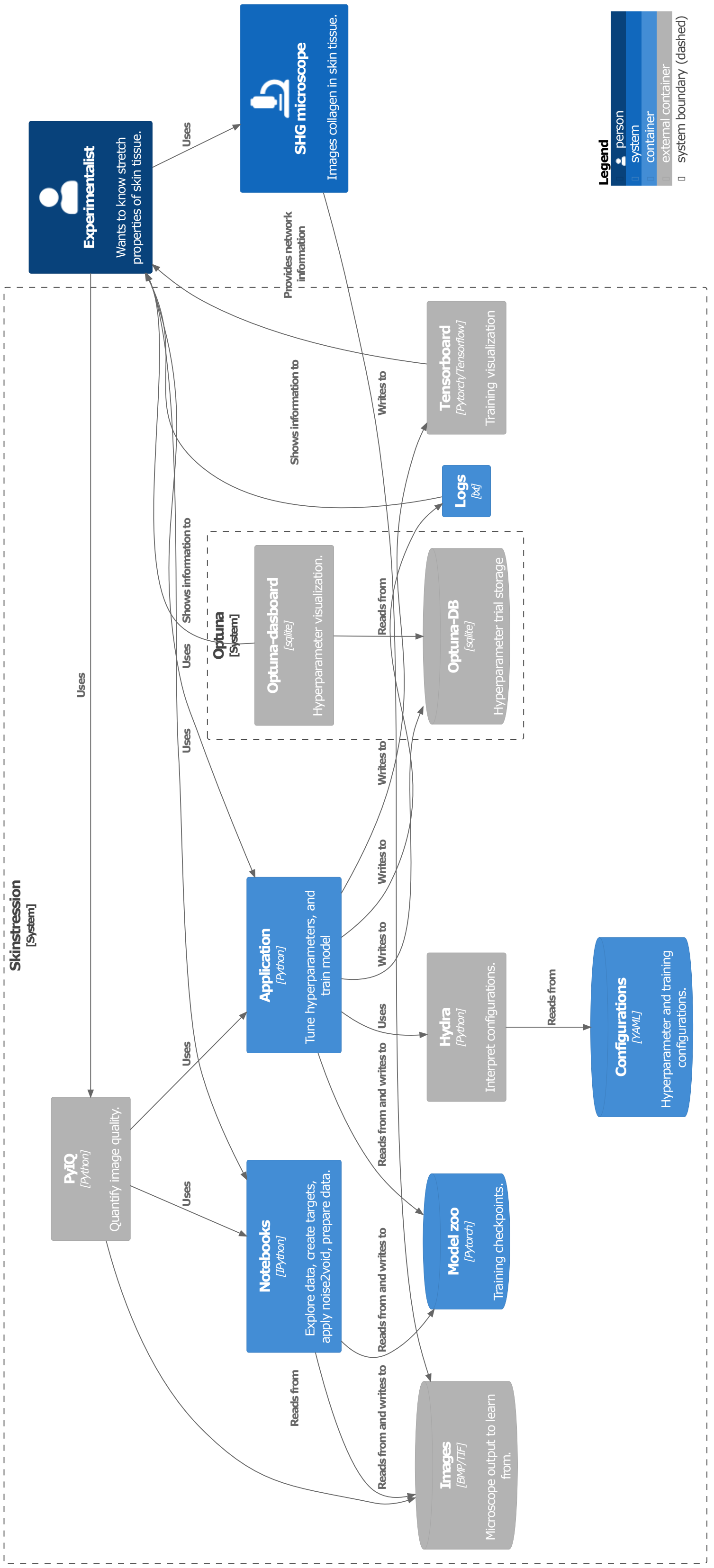


Figure A.5: Container diagram of Skinstression. The bounding box shows internal communications of Skinstression. Images generated with the SHG microscope get stored and can be read by PyimageQualityRanking (PyIQ). PyIQ sorts the images by quality, such that they can be read in order by notebooks and the main application. The main application reads locally stored configurations using Hydra. Trained models are stored in the model zoo. Hyperparameter optimizations are tracked by Optuna and stored to an SQLITE database. The Optuna database can be inspected by Optuna-dashboard. Training and hyperparameter optimization can also be inspected by Tensorboard. The application logs output and errors to text files.

B

SCLICOM

B.1 Derivation of ROC and PR curve baselines

B.1.1 ROC curve baseline

Given that accuracy

$$\mathrm{acc} = \pi \cdot \mathrm{TPR} + (1-\pi)(1-\mathrm{FPR}), \tag{B.1}$$

which can be rewritten to TPR in terms of FPR,

$$\begin{aligned}\mathrm{TPR} &= \frac{\mathrm{acc}}{\pi} - \frac{(1-\pi)}{\pi}(1-\mathrm{FPR}) && \text{(B.2)}\\ &= \frac{\mathrm{acc}}{\pi} - \frac{(1-\pi)}{\pi} + \frac{1-\pi}{\pi}\mathrm{FPR} && \text{(B.3)}\\ &= \frac{\mathrm{acc}-1+\pi}{\pi} + \frac{1-\pi}{\pi}\mathrm{FPR}. && \text{(B.4)}\end{aligned}$$

In the case of an always positive classifier that is right for π of the time,

$$\mathrm{TPR}(\mathrm{acc}=\pi) = \frac{1-\pi}{\pi}\mathrm{FPR} + 2 - \frac{1}{\pi}. \tag{B.5}$$

B.1.2 PR curve baseline

Given that the F_1-score is

$$F_1 = \frac{2}{\mathrm{prec}^{-1} + \mathrm{rec}^{-1}}, \tag{B.6}$$

which can be rewritten as

$$\mathrm{prec} = \frac{F_1 \cdot \mathrm{rec}}{2 \cdot \mathrm{rec} - F_1}. \tag{B.7}$$

The F_1-score of an always positive classifier is

$$F_{1,+} = \frac{2 \cdot \mathrm{prec}}{\mathrm{prec} + 1}, \tag{B.8}$$

since in this case prec $= \pi$ and rec $= 1$. Substituting F_1 in Equation B.7 for $F_{1,+}$, we get

$$\mathrm{prec} = \frac{F_{1,+} \cdot \mathrm{rec}}{2 \cdot \mathrm{rec} - F_{1,+}}. \tag{B.9}$$

B.2 Flow of images to splits

The splits are created as described in Subsection 4.3.7. The process is visualized in Figure B.1.

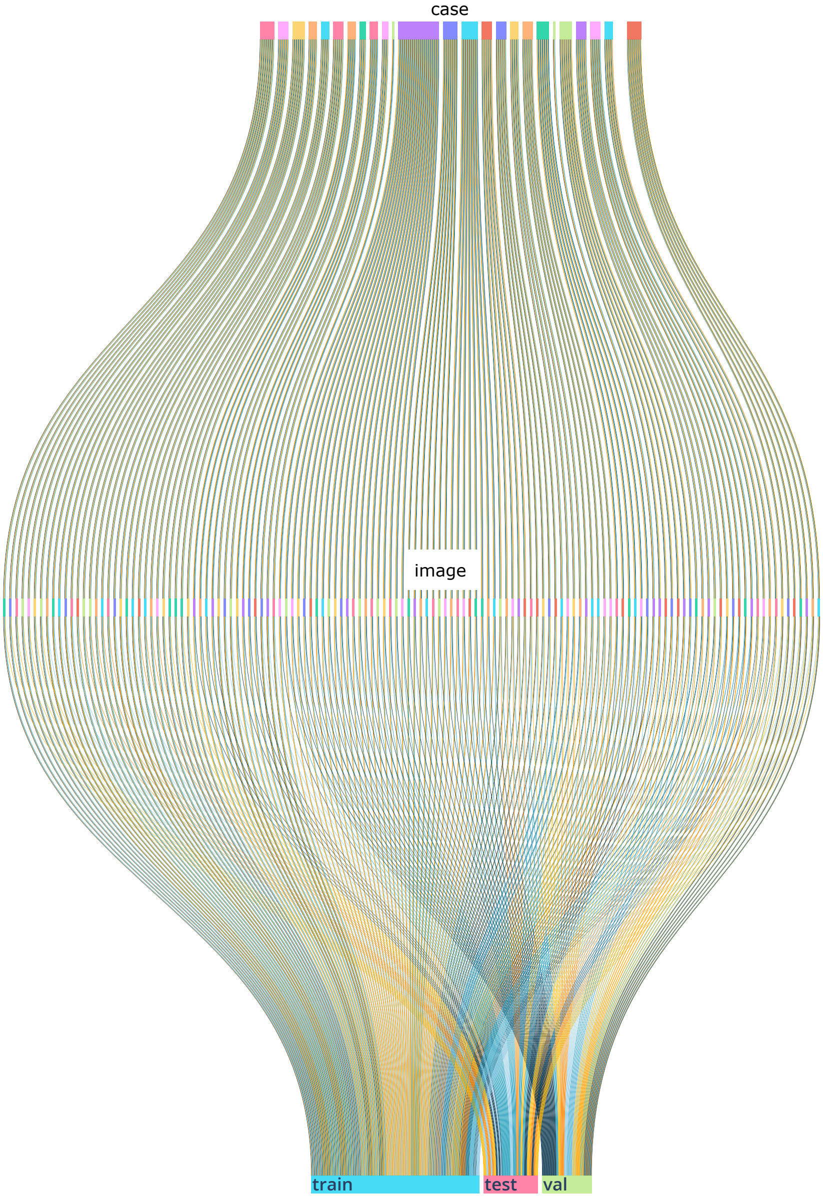


Figure B.1: The flow of images to splits. Top row shows available cases. Every block is one case. Middle row shows available images and is linked to the cases. Bottom row shows training, validation and test splits. Colors show flow within one fold. Manual zoom may show details.

B.3 Software diagrams

The first level of the C4 model sets the context of the software and is shown in Figure B.2. The second level shows the high-level technical building blocks of the software and is shown in Figure B.3.

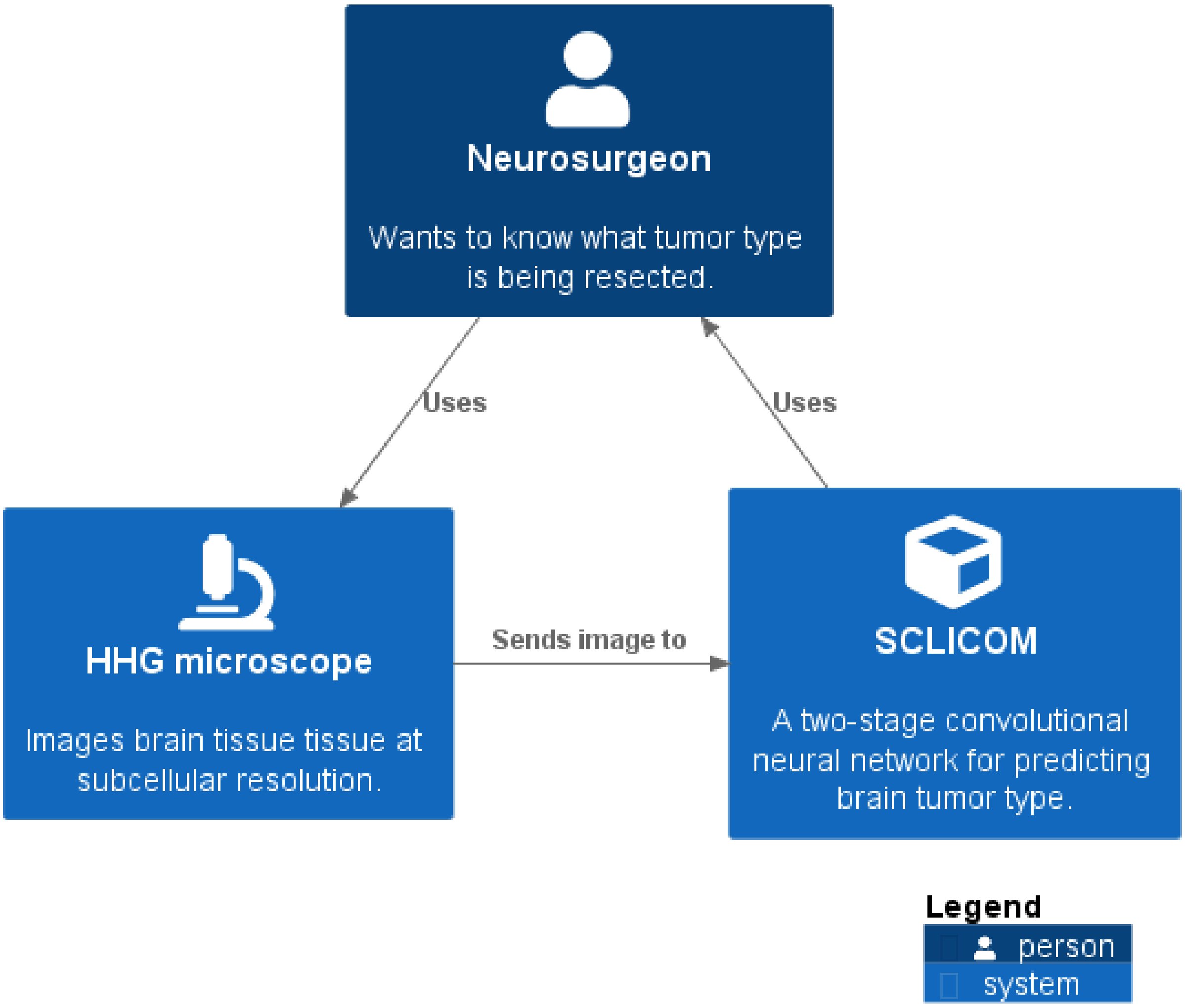

Neurosurgeon
Wants to know what tumor type is being resected.
Uses
Uses
HHG microscope
Images brain tissue tissue at subcellular resolution.
Sends image to
SCLICOM
A two-stage convolutional neural network for predicting brain tumor type.
Legend
person
system

Figure B.2: System context diagram of SCLICOM. A neurosurgeon or technician images brain tumors using an HHG microscope. The microscope output serves as input to SCLICOM, a convolutional neural network to predict brain tumors imaged by the microscope. The trained model can help intraoperatively diagnose brain tumors, or provide specific regions to at for quicker diagnosis.

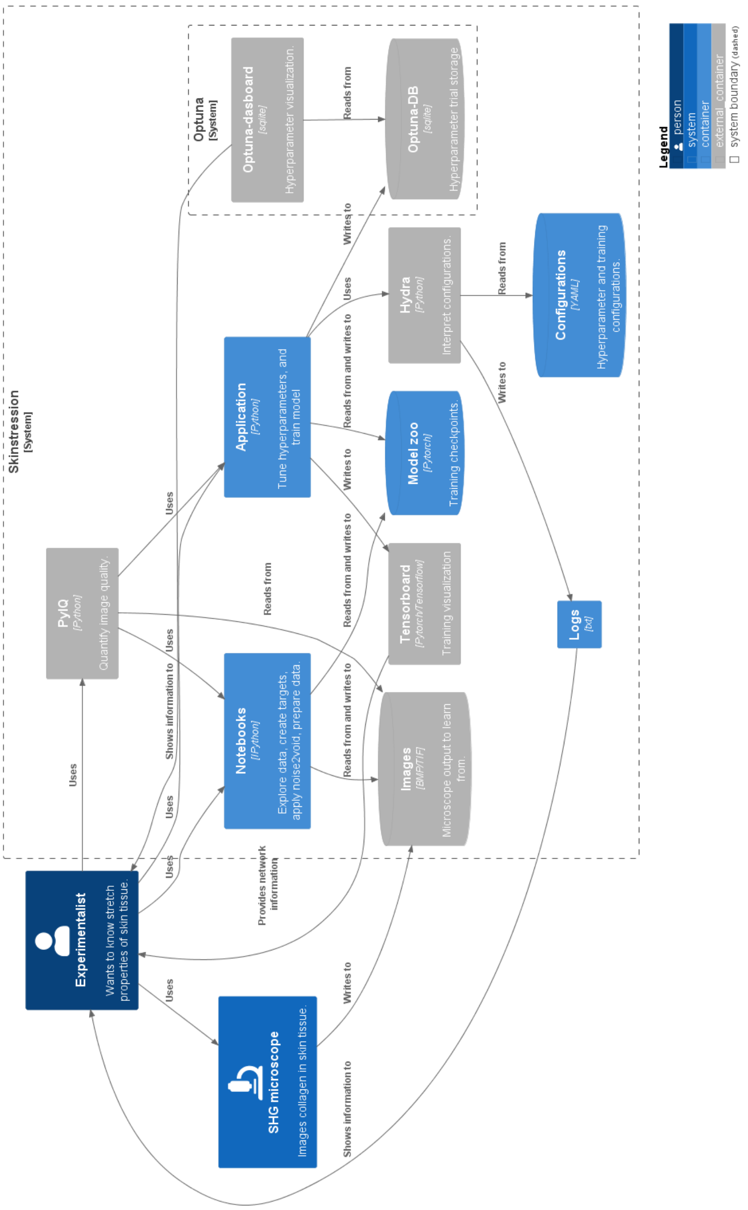


Figure B.3: Container diagram of SCLICOM. The bounding box shows internal communications of SCLICOM. Images generated with the HHG microscope get stored. The main application reads locally stored configurations and trains with Pytorch Lightning. Trained models are stored in the model zoo. Hyperparameter optimizations and resources are tracked by Ray Tune. Training and hyperparameter optimization can be inspected by Tensorboard. The application logs output and errors to text files.