Computing and Informatics, Vol. 39, 2020, 780–807, doi: 10.31577/cai_2020_4_780

BREAST HISTOPATHOLOGY WITH HIGH-PERFORMANCE COMPUTING AND DEEP LEARNING

Mara Graziani

University of Applied Sciences of Western Switzerland
HES-SO Valais, Rue de Technopole 3
3960 Sierre, Switzerland
&
Department of Computer Science, University of Geneva
Battelle Building A, 7, Route de Drize
1227 Carouge, Switzerland
e-mail: mara.graziani@hevs.ch

Ivan Eggel

University of Applied Sciences of Western Switzerland
HES-SO Valais, Rue de Technopole 3
3960 Sierre, Switzerland
e-mail: ivan.eggel@hevs.ch

François Deligand

INP-ENSEEIHT
2 Rue Charles Camichel
31000, Toulouse, France
e-mail: francois.deligand@laposte.net

Martin Bobák

Institute of Informatics
Slovak Academy of Sciences
Dúbravská cesta 9, 845 07 Bratislava, Slovakia
e-mail: martin.bobak@savba.sk

Vincent Andrearczyk

University of Applied Sciences of Western Switzerland
HES-SO Valais, Rue de Technopole 3
3960 Sierre, Switzerland
e-mail: vincent.andrearczyk@hevs.ch

Henning Müller

University of Applied Sciences of Western Switzerland
HES-SO Valais, Rue de Technopole 3
3960 Sierre, Switzerland
&
Radiology Service, Medical Faculty, University of Geneva
Geneva, Switzerland
e-mail: henning.mueller@hevs.ch

Abstract. The increasingly intensive collection of digitalized images of tumor tissue over the last decade made histopathology a demanding application in terms of computational and storage resources. With images containing billions of pixels, the need for optimizing and adapting histopathology to large-scale data analysis is compelling. This paper presents a modular pipeline with three independent layers for the detection of tumoros regions in digital specimens of breast lymph nodes with deep learning models. Our pipeline can be deployed either on local machines or high-performance computing resources with a containerized approach. The need for expertise in high-performance computing is removed by the self-sufficient structure of Docker containers, whereas a large possibility for customization is left in terms of deep learning models and hyperparameters optimization. We show that by deploying the software layers in different infrastructures we optimize both the data preprocessing and the network training times, further increasing the scalability of the application to datasets of approximatively 43 million images. The code is open source and available on Github.

Keywords: Histopathology, exascale, medical imaging, sampling

1 INTRODUCTION

Breast cancer is the second leading cause of cancer death among women worldwide [35]. In 2019, the estimated number of women diagnosed with breast cancer was 271 270 only in the U.S. (7.3 % increase from the estimates of 2017), and an in-

creasing number of women died from the disease (2.8 % increase from 2017 with 41 760 estimated deaths). A metastasizing breast cancer, particularly, has a far worse prognosis than a localized one. Assessing regional tumor spreading is thus extremely important for prompt treatment planning and accurate cancer staging [12]. Being the most likely target for initial metastases, axillary lymph nodes are analyzed to determine the spreading stage to neighboring areas. The N-stage of the TNM system, for instance, assesses the presence of tumor in regional lymph nodes. Before undergoing surgical removal of tissue, the patient is injected a blue dye or a radioactive tracer to identify the nearest lymph node to which the tumor may have drained, which is also called the sentinel lymph node [12]. Thin tissue slices are collected for visual analysis, mounted on glass slides. At this stage, the tissue slices mostly have transparent cells, the reason why they are treated with multiple contrasting stains. Different staining techniques can be used, with Hematoxylin and Eosin staining (H & E) being the most common, and immunohistochemical (IHC) staining for cytokeratin being used only in case of unclear diagnosis on H & E [7]. Hematoxylin stains the nuclei with blue, while Eosin highlights the cytoplasmic and non-nuclear components in different shades of pink. Their combination highlights the structure and cells within the tissue specimen. Tumor presence is traditionally evaluated by microscopic inspection, which may take several minutes per slide. Pathologists base their decisions upon morphometric and architectural features of the tissue structure and the nuclei, for instance estimating the presence of tubular formation, nuclear pleomorphism and mitotic count (see Figure 1) [30]. Such analysis is time-consuming, tedious, and error-prone since small metastases may be missed by the pathologist's eyes (detection rate lower than 40% for the smallest type) [38]. Moreover, the grading of tumor growth patterns is very subjective and reports high inter-observer variability ($\kappa \in [0.582, 0.850]$ [32, 29]).

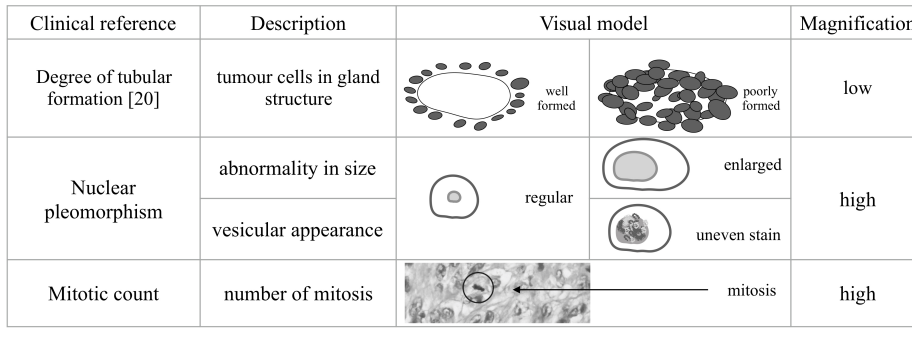


Figure 1. Grading criteria in the Nottingham Histologic Grade of breast cancer

The difficulties in the diagnostic motivate the automation of the grading process, made possible by the commercialization of high-resolution digital scanners for images of tissue specimens called Whole Slide Images (WSIs) around the year 2000 [1]. The well-established success of Convolutional Neural Networks (CNNs) in various

computer vision tasks triggered their use for automatic diagnoses. While clinical pipelines are increasingly introducing the full digitization of specimens by WSIs, several challenges affect the analysis of these images by deep learning models. On the data acquisition side, defects in the tissue handling, fixation, processing and staining affect the image quality, thus hampering future analyses. Intervals between tissue harvest, fixation and total fixation time are also poorly controlled in current pipelines, often leading to high variability and heterogeneity in the data collection [1]. Further technical aspects of the slide digitization such as the maximum magnification, the image compression and the color palette may also cause heterogeneity in the data, affecting both the image visualization and analysis. Moreover, storage requirements can easily explode when scaling the data collection to multiple patients. WSIs are extremely large, reaching generally more than $100\,000 \times 100\,000$ pixels [4]. Some pathological section processes may generate up to 20 images per patient, easily leading to more than 65 GB of data per patient [41]. Besides, no universally accepted WSI format has yet been defined, with different scanner manufacturers patenting different WSI file formats (such as SVS or NDPI). The multiresolution pyramidal structure of WSIs, finally, contains multiple down-sampled versions of the original image, with different informative content [4]. Varying the scale at which the images are analyzed improves the quality of the analysis by taking into account information at different levels, the high-level disposition of the cells (e.g. degree of tubular formation) and fine-grain details such as the nuclei morphology or the mitotic activity.

This paper describes the research pipeline that we developed to process a scalable number of breast lymph node WSIs, to compare the performances of multiple deep learning architectures in terms of their training and inference time, accuracy and explainability. Our design takes into account the developmental requirements of scientific research generally performed on small computing clusters (up to 4 GPUs maximum). Being released with a containerized approach, it can be deployed on different infrastructures without requiring specific expertise, and it can be used to run some tasks on the large scale computing services provided by the PROCESS project[1], by transforming the Docker into Singularity containers. The design of such pipeline was driven by the need for flexibility in the used libraries for scientific development, while its integration exploits the services of the High-Performance Computing (HPC) structures. The paper is structured as follows. Section 2 introduces the state of the art about deep learning developments for both digital pathology and HPC. Section 3 describes the methods and datasets used for the experiments. In this section, we also describe the specific requirements of deep learning for histopathology and how these are met by the PROCESS platform interconnecting HPC and research centers in Europe. The experiments are run, in fact, at different sites, namely the University of Amsterdam (UVA), the SurfSARA computing center (with the LISA cluster), the SuperMUC-NG of the Leibniz-Rezerchcentrum (LRZ), the Prometheus cluster

[1] The PROCESS project received funding from the European Union's Horizon 2020. Homepage: `https://www.process-project.eu/`

at Cyfronet (AGH), the computing resources of the Slovak Academy of Sciences (UISAV) and our local resources. Particularly in Sections 3.5, 3.6 and 3.7, we describe the three layers of the proposed pipeline, pointing to the relative open source Github repositories. Section 4 reports the experimental results obtained with the proposed architecture on the different computing resources. In Section 5, finally, we present a discussion of the results and the challenges of scaling histopathology analysis to exascale datasets, together with the computational aspects of the software optimization.

2 RELATED WORK

2.1 Digital Pathology

Digital pathology has become very popular over the last decade for its practical assets [26]. WSIs reduce the need for storing glass slides on-site, reducing their risk of breaking, fading, or getting lost. Digital images can also easily be shared across institutions. This solicits the exchange of opinions, and if necessary pathology consults, about challenging cases. In case of specific questions at tumor boards, WSIs can be inspected offhand to find answers. Besides, WSIs allow the collection of examples of many different cases, including rare cases, that can be practically used in teaching slides sets. The creation of open-access digital slide archives, such as that of The Cancer Genome Atlas, led to intensive research on image analysis for pathology, which recently steered towards the application of deep learning based techniques. The large image sizes, as well as their heterogeneity and the often-imbalanced data distribution (towards non-tumorous tissue), make the development in this area challenging. The development of handcrafted features requires a background knowledge comprehensive of biology and imaging skills, together with the further specialization of the specific tissue type being analyzed. Handcrafted features that can be applied to any organ type include nuclear/gland shape and size, and tissue texture and architecture. Nuclear shape and texture, for example, were shown to predict the prognostics for breast cancer [25]. Graph-based approaches such as Voronoi tessellations were also used to characterize the relative positioning of nuclei and glands within the tissue. Prognostic features specific for breast cancer, for example, take into account the disposition of infiltrating tumoral lymphocytes.

Deep learning approaches for the analysis of histopathology images remove the need for the cumbersome creation of tissue-specific handcrafted features. Most of the approaches in the literature focus on the detection of Region of Interest (ROIs) where tumorous tissue can be identified, on the direct segmentation of nuclei [19] or the quantification of mitoses [3]. Because of the pyramidal structure of WSIs, images are often analyzed at multiple resolutions in a multi-step procedure [22, 42]. A thresholding operation (e.g. Otsu thresholding [42]) is performed on the lowest resolution image to filter out the image background. The doctor annotations of ROIs can then be used as ground-truth labels to train Convolu-

tional Neural Networks (CNNs). Image crops at higher resolutions are assigned the tumor label if they fall within the ROI while the remaining tissue is assigned a non-tumor label. These data are used to train CNNs end-to-end for the patch-based classification of tumorous images[2]. Additionally to these, weakly supervised learning methods were used to exploit coarse labels to extract fine-grained information. The information contained in high-resolution crops of the WSIs is used as a "bag of inputs" with a single label in multiple-instance learning, for example. Teacher-student designs, besides, were recently proposed for combining small amounts of finely annotated data (manual segmentations of tumorous areas) with large amounts of weakly annotated data such as instance-level annotated WSIs [28].

As for the deployment of digital pathology on HPC, this is a recently growing area with yet unexplored opportunities and challenges. Particularly, with the exponential growth of biomedical data, several techniques will require adaptation and optimization to process efficiently large-scale datasets [41]. The specific demands of digital pathology require double adaptation, namely that of the HPC infrastructure towards the high computational burden of analyzing the massive dataset sizes, and that of developing pipelines that best exploit the infrastructure potential. Only a few existing works distribute the algorithms for medical image analysis. The work in [39], for example, uses MapReduce to answer spatial queries on large scale pathology images. A low-cost computing facility that can support the analysis of up to 1 million (1 M) images was proposed, for example, in [5]. Multiple-instance learning for histopathology was implemented in [41] to fit Microsoft HPC resources, and further modifications to the original algorithm were shown to obtain better scalability in [40]. These works, however, were directly optimized on the available computing resources for the research, requiring expertise at the frontier of HPC, deep learning and digital pathology. The main purpose of this work is to offer a ready-to-use application for researchers in the digital pathology field that have little to no experience in HPC and optimized computing. The application, being organized in three layers, presents different steps in the traditional pipeline with a modular approach, where each module can be customized in terms of research parameters and input data and can be deployed to different computing facilities.

2.2 HPC for Deep Learning

HPC infrastructures have played a fundamental part in solving large-scale problems with a high degree of computational complexity in domains such as astrophysics, high energy physics, computational fluid dynamics or finite element analysis. HPC services were built specifically to fit the requirements of such problems to efficiently scale up the problem size, exploiting the power of thousands of multi-core computer

[2] Different pipelines can be adopted, obtaining different results, as those in `https://camelyon17.grand-challenge.org/evaluation/leaderboard/`

nodes to effectively solve a single computationally-intensive problem. The downside is that the technical and organizational architecture of HPC trades some of the flexibility and dynamism for achieving the highest possible performance and maximum utilization rate. Only recently, HPC has embraced some of the programming solutions to provide more effective Single Instruction Multiple Data (SIMD) operations for vector data, which generated the possibility of introducing large scale machine learning applications to their computational capacities.

The training of a deep neural network involves solving a high-dimensional non-linear optimization function. The minimization of the gradients, often performed with gradient descent algorithms such as Stochastic Gradient Descent (SGD), uses several linear algebra computations on generally dense matrices. DistBelief was one of the first examples of parallel training frameworks to train fully connected networks (42 M parameters) and locally connected convolutional neural networks (17 M parameters) [9]. In this case, the parallelization of network training was achieved by *model parallellism*. Model parallellism, splits the weights of the network equally among multiple threads, creating network blocks that all work on the same mini-batch. In other words, the same data is used for every thread, but the model-inherent computations are split among threads. The output needs therefore to be synchronized after each layer to obtain the input to the next layer. A simpler method is that of *data parallelism*. The same model is replicated in each thread or working node (either GPU or CPU) and the training is performed on different batches of data. The gradients computed by each thread are relevant to the overall network training. Hence, they need to be shared within all the models on all the workers at the end of each data pass (i.e. their average is computed). Despite the simplicity and the adaptability of this method to datasets, models and resources, two possible bottlenecks may arise and hamper its efficiency. The averaging of the model parameters would require the transmission of extremely large matrices between nodes, thus requiring a highly efficient network card connecting each node. A similar problem arises when handling the multiple data accesses: the continuous input/output operations and data decoding may be the first cause for the slowing down of the training process. Moreover, as in data parallelism the same mini-batch is analyzed by all the GPUs, smaller batches result in decreased efficiency. The work on Large Minibatch SGD by [13] proposed a solution to the optimization difficulties that arise when increasing the size of the mini-batches to push to the edges the computational gains in each worker. Furthermore, recent research in gradient compression has proposed a solution to the latency in the communication between different nodes caused by limited communication bandwidth. For instance, gradient compression transmits only gradients larger than a certain threshold and accumulates locally the rest of the gradients, thus consistently reducing the network communication time [23].

In both data and model parallelism, the communication between the nodes can be either synchronous or asynchronous. In the former, all the devices use different parts of the same batch of training data and the model is updated when the computation has finished in all of them. In the latter, the devices update the

model independently based on their mini-batches. Generally, the updates are shared through a central parameter store, called parameter server. The parameter server receives the gradients from all the workers and, when all the updates have been received, computes the new model update and sends it to the workers. A boost in efficiency is given by the ring all-reduce technique, where each worker receives the latest gradient update from its predecessor and sends its gradients to its successor neighbor in a circular fashion. The trade-off between the synchronous and asynchronous implementation of SGD was exploited in [21]. Synchronous systems use the hardware resources less efficiently, whereas the asynchronous systems generally need more iterations since the gradient updates are computed on older versions of the model.

The use of specific hardware dedicated to deep learning seems, therefore, to be projected as a prosperous newborn branch for HPC. The introduction of the ultimate TPUs by Google research[3] stands as an initial step in this direction. Notwithstanding, the range of hardware characteristics of multi-purpose supercomputers is very large. Scaling up computations might be cumbersome, requiring HPC expertise to tailor models on the available system hardware.

3 DATASETS AND METHODS

3.1 Datasets

We use the Camelyon 16 and 17 challenge data, which constitute, currently, one of the largest and most challenging datasets for histopathology research [4]. The datasets include WSIs of lymph node sections together with slide-level annotations of metastases type (negative, macro-metastases, micro-metastases, isolated tumor cells) and some manual segmentations of tumor regions. The data were collected at five different data centers, namely the University Medical Center in Nijmegen (RUMC), the Canisius-Wilhelmina Hospital in Nijmegen (CWZ), the University Medical Center Utrecht (UMCU), the Rijnstate Hospital in Arnhem (RST), and the Laboratory of Pathology East-Netherlands in Hengelo (LPON). A summary of the data provenances and distribution is given in Table 1.

The variability in preparation across acquisition centers makes the data very heterogeneous. Three different scanners were used in the five centers, namely the 3DHistech P250 (0.24 μm pixel size) at RUMC, CWZ and RST, the Philips IntelliSite Ultra Fast Scanner (0.25 μm pixel size) at LPON and the Hamamatsu XR C12000 (0.23 μm pixel size) at UMCU. The average file size is around 4 GB, for a total storage requirement of 3 030.5 GB.

[3] `https://cloud.google.com/tpu/docs/tpus`

Year	Center	Total WSIs		Metastases (Train)			
		Train WSIs	Test WSIs	None	ITC	Micro	Macro
2016	RUMC	170	79	100	–	35	35
	UMCU	100	50	30	–	12	8
2017	CWZ	100	100	64	11	10	15
	RST	100	100	58	7	23	12
	UMCU	250	100	165	2	34	49
	RUMC	349	100	210	8	64	67
	LPON	100	100	61	8	5	26
Total		1 169	629	688	36	183	212

Table 1. WSI-level summary of the Camelyon 16 and 17 challenge datasets

3.2 Application-Driven Requirements

The design of the HPC infrastracture used to run the experiments was driven by the specific requirements of the pathology application, summarized in Table 2. The rapid growth of this field makes its requirements at the border of those of exascale computing both for computational and storage resources.

On the storage side, the extraction of image crops from the WSIs can easily grow to more than 60 thousand patches for a single record. With more than one record being held for each patient, the storage requirements can easily grow to several Terabytes (TB) of space. This highly demanding data preprocessing is shared with similar data types, e.g. satellite images. A requirement that is specific to the medical field, however, is the handling of sensible data. Large publicly available datasets such as Camelyon can be shared and downloaded on the main infrastructure to exploit the storage and computing facilities of the HPC providers. Sensible data such as private patient data must, however, be left on the local storage of the hospital to meet the security and privacy requirements. A typical approach, in this case, is that of the "Evaluation as a Service" (EaaS) solution, where the data can remain in a single infrastructure and does not need to be moved. Sensitive data could be left in the hospitals and be used only on their local environment.

On the computational side, the training of state-of-the-art CNNs with millions of parameters is highly demanding. Current implementations can take days to converge for datasets sizes of the order of magnitude of 10 Gigabytes (GB). Medical imaging data (and not only) easily reaches the TB if not the petabyte (PB) order of magnitude. The computational demand on such datasets can easily reach 15 petaflop/s [21], pointing towards exaflop/s in the nearest future. The support of dense linear algebra on distributed-memory HPC is an indispensable requirement to train deep models. Open Message Passing Interfaces and parallelization libraries such as Horovod[4] allow the parallelization on multiple GPUs and HPC nodes. Top-development libraries such as Tensorflow and Keras are also top list

[4] `https://eng.uber.com/horovod/`

requirements for the deployment of deep learning algorithms. Computational and storage requirements may seem two disentangled types of prerequisites, but one actually proportionally influences the other. With increasingly intense computations, the need for storage for saving intermediate results arises.

The need for containerized software is a further requirement to maintain the portability of the developments. While Docker containers are mostly used by the scientific community, Singularity containers constitute a better trade-off between the application and the infrastructure requirements, providing improved security by removing the root access on the HPC machines. Specific technologies are required to process the different image formats, being WSIs often saved as BIGTIFF files paired to annotation XMLs, CSVs or TXTs. Moreover, datasets such as PubMed central may require the handling of more common image compression formats such as JPEG and MPEG.

Requirement	**Motivation**	**Software Layer**
SCP and FTP connection	initial WSI transfer	1
Docker or Singularity	software portability	1, 2, 3
Data storage (> 100 TB)	public WSIs and intermediate results	1, 2
OpenSlides	WSI preprocessing	1
Tensorflow $> 1.4.0$	DL library	2, 3
Keras $> 1.4.0$	DL library	2, 3
Horovod	distributed computing	1, 2
OpenMPI	distributed computing	1, 2
SLURM	distributed computing	1

Table 2. Summary of application specific requirements

3.3 The Exascale Platform Architecture

In this section, we describe the architecture of the exascale platform within the developments for the PROCESS project, highlighting the modules introduced to adapt High-Performance Computing (HPC) to the use case requirements.

Data Services for Medical Use Case. The main data service of the PROCESS platform is a virtualized distributed file system (VDFS) LOBCDER. It is a modular, scalable and extensible VDFS implementing the micro-infrastructure approach. LOBCDER integrates special hardware nodes that are dedicated to the transfer of data. It has a module for meta-data management (DataNet) and pre/post-processing of exascale datasets (DISPEL).

Computing Services for the Medical Use Case. The PROCESS platform is available via Interactive Execution Environment (IEE) which gives access to heterogeneous computing infrastructure of supercomputers supporting HPC (via Rimrock) as well as cloud computing (via Cloudify). Both computing services provide relevant containerization services (Docker and Singularity). The

platform also allows users to use GPUs in parallel and in a distributed manner.

Table 3 summarizes the core requirements for the exascale platform given by the medical application. These requirements are one of the main pillars on which the PROCESS architecture was built. The table also provides an overview of how the PROCESS platform is capable to satisfy them.

Requirement	**PROCESS Module**
Support containerization	Rimrock
Workflow management for configuration, deployment and management of multiple application/use case executions	IEE
Distributed file system supporting medical use case datasets and their file formats (e.g. image formats)	LOBCDER
Support multiple pipelines within a workflow	LOBCDER, Rimrock
Distributed pre-processing of training data	DISPEL, LOBCDER
Supporting data transfer protocols (e.g. GridFTP, SCP, etc.)	LOBCDER
Support of the set of common tools for machine learning and deep learning on the HPC centres	Docker and Singularity containers supported by computing sites
Parallel and distributed multi-GPUs training	Computing centres
Support GPUs in containers	Computing cetenrs

Table 3. Use case requirements and dedicated components from the PROCESS platform which fulfils them

3.4 A Modular Design for a Step-by-Step Pipeline

The method we propose is a three-layer software architecture for training different deep neural network models, summarized in Figure 2 and presented in detail in Figure 3. The modular approach splits the full pipeline in different workflows that can be shared, reused and updated independently. In the following subsections we describe the methods adopted in each of the three framework layers.

3.5 Layer 1: Preprocessing and Patch Extraction

The WSI in its original format is much larger than the maximum input format for a CNN. For this reason, WSIs need to be cropped into smaller images, called patches or tiles, which can then be fed to the network. Patches are extracted at the highest level of magnification (i.e. 40×). High resolution highlights qualitative features of the nuclei which are prognostic of cancer [30].

The first layer of the proposed application focuses on the patch extraction and data preprocessing of the WSIs (see Figure 4). As a first step, the filesystem is

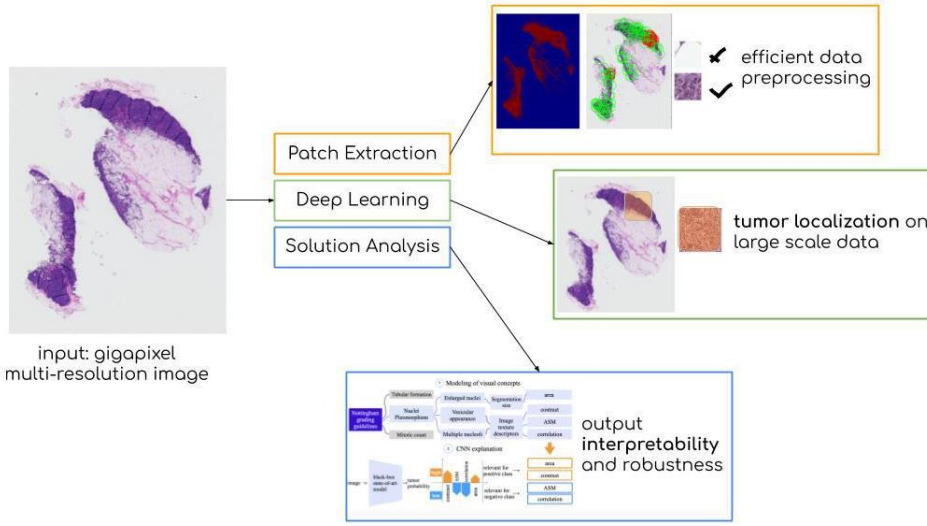


Figure 2. Overview of the CamNet software

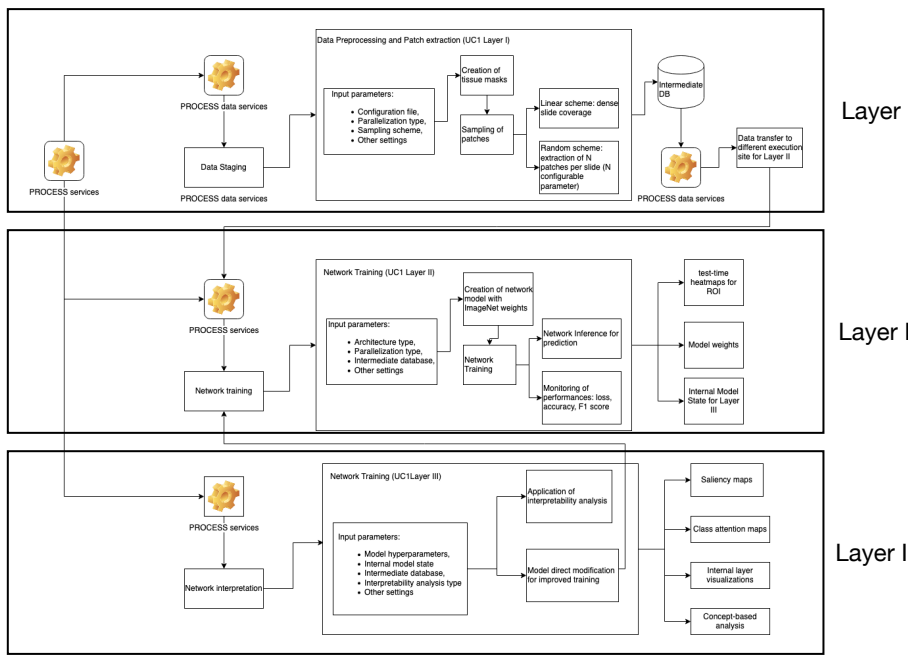


Figure 3. Detailed structure of each software layer. Best on screen.

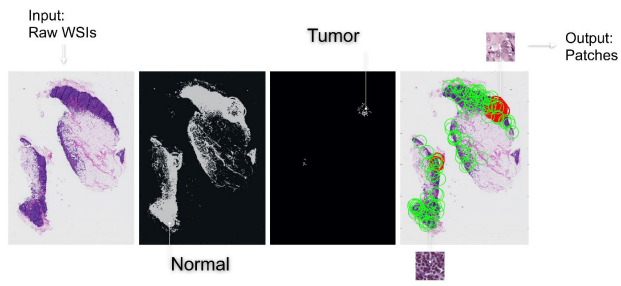


Figure 4. WSIs preprocessing pipeline: Normal tissue and tumor tissue masks are extracted and high-resolution patches are sampled from the selected regions

scanned to retrieve patient-related metadata and the acquisition center. If additional data were to be loaded on the HPC, the LOBCDER filesystem of PROCESS could be used. From the lowest magnification level, binary masks of normal and tumor tissue are extracted by the Otsu thresholding method [42], already proposed for this data in [4]. Patches are then sampled at the highest magnification level from the annotated tumor regions for the tumor class. For the non-tumor class, patches are sampled not only from the annotated images but also from 297 non-tumor WSIs in Camelyon17.

Patches with non-relevant information (e.g. white content, black pixels, background, etc.) are filtered out and discarded. Information about the patient, the lymph node, the hospital which handled the acquisitions, the resolution level of the patch and the patch location in the WSIs are stored together with the pixel values in an intermediate HDF5 database. Moreover, the doctor annotations are stored in the HDF5 as a binary label on the patch, which discriminates between tumor and non-tumor patches. Different cropping strategies following the sampling approaches for automated histological image analysis [4] are available in the system. The most common is the random sampling scheme, which extracts randomly several patch locations. A seed for initializing the random sampling and the desired number of patches to extract (N) are passed as input parameters of this stage. A white-threshold (expressed in terms of the percentage of pixels with intensity values larger than 200) is applied to discard image croppings of the background and the adipose tissue, which are uninformative for the task. Similarly, black background patches are removed from the sampling results. The Simple Linux Utility for Resource Management system (SLURM) is used to develop a parallel version that can be distributed on the HPC. A job array is configured with a random generator seed. At each batch run, a different patch set is extracted. The code for local execution (sequential algorithm) and for the distributed execution is available online[5].

[5] `https://github.com/medgift/PROCESS_L1`

By means of the HPC computational capacity, the dense coverage of the WSIs is also possible. A sliding window that extracts patches with a fixed stride is implemented as an alternative sampling option. This option, not possible with the local research facilities, is optimal when ran on the computing capabilities of HPC. The distribution of the code on different HPC nodes is, in this case, necessary. Therefore, this part of the software can only be run on distributed computing facilities with the support of SLURM. Each WSI is assigned to a single SLURM job array. The patches with non-relevant information are filtered out and discarded by the white thresholding. A further scaling step is also available, combining two SLURM techniques at the same time for better efficiency. Each task in the SLURM job array is assigned a different WSI, and an arbitrary number of subtasks are run in parallel to implement the sliding window.

The image croppings resulting from the extraction process with any of the two strategies are stored in an intermediate Hierarchical Data Format file (HDF5), together with the metadata about the patient, the lymph node, the acquisition center, the magnification level and the patch location.

3.6 Layer 2: Patch-Based Deep Learning

The second layer loads the intermediate HDF5 dataset generated by Layer 1, and focuses on the training of deep learning architectures for the binary classification between tumor and non-tumor patches, following the approach in [42]. The training of several state-of-the-art CNNs (i.e. ResNet50, ResNet101, GoogleNet, Inception V3, DenseNet, InceptionResNetV2, all pre-trained on ImageNet) is pre-implemented. The training can be distributed with the Open Source Distributed Deep Learning Framework for Tensorflow, Horovod. A configuration file is used to specify the network architecture preferences and hyperparameters (i.e. loss, activation, learning rate and batch size). The output of this layer consists of the trained network parameters and training statistics. In this paper, we compare the performances of ResNet and Inception on different GPU types, namely Titan V, Titan X, Tesla K80, Tesla K40. The source code is also available online, for either single-GPU or multi-GPU execution[6].

3.7 Layer 3: Inference and Interpretability

We present a summary of the functionalities of Layer III in Figure 5. This last layer of the proposed application deals with two main tasks: performing inference and interpreting the models trained in Layer II.

As a first functionality, this layer generates heatmaps of the probability of the presence of tumorous tissue, which can be used for visual inspection and comparison with the ground truth segmentations. CNN inference, in general, is less costly than training, although WSIs still require a great number of patches to be tested. This

[6] `https://github.com/medgift/PROCESS_L2/`

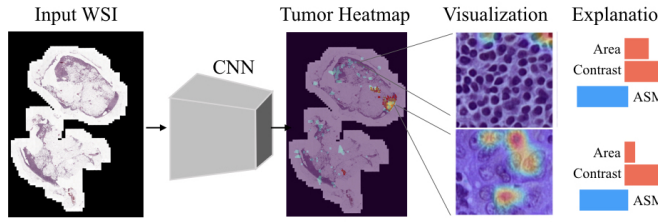


Figure 5. An example of the functionalities in Layer III, which consist of performing distributed inference for building the tumor heatmap and providing interpretability analyses and insights about network training

process is hence optimized by parallel-data distribution. Several copies of the CNN are stored on the GPUs and inference is performed in parallel on distinct batches of data. The heatmaps are then built by interpolating the predicted probability values for each pixel.

The interpretability of the models, besides, is analyzed in this layer. Understanding the decision-making process of CNNs is a key point in medical imaging, to ensure that clinically correct decisions are taken. Several different approaches are proposed in the literature, with clear distinctions being made between models that introduce interpretability as a built-in additional task [11, 20, 6, 8, 2, 34] and post-hoc methods. Post-hoc methods, as defined in [24], are particularly suited to the proposed layered framework, since they allow to disentangle the interpretability analysis from network training. They can be used to explain any machine learning algorithm without retraining the network weights. Several post-hoc techniques [36, 43, 33, 10, 31] highlight the most influential set of features in the input space, a technique known as attribution to features [37]. Among these, gradient-based approaches, such as Class Activation Maps (CAM), Gradient-weighted Class Activation Maps (grad-CAM) and its improved version grad-CAM++, attribute the network output to perturbations of the input pixels. The output of such methods is a heatmap of the input pixels that mostly contributed to the decision. Local Interpretable Model-Agnostic Explanations (LIME) are used as an alternative tool to obtain visualizations. These visualizations are compared to interpreting the CNNs by the regression of clinical values such as lesion extension and appearance in the internal network activations in [15]. The method proposed for the comparison is that of Regression Concept Vectors (RCVs), first implemented in [14] and then expanded in [17]. This approach, besides, was further investigated in [16], showing that it can be efficiently used to improve the training of CNNs on medical images. To develop concept-based explanations, we define a list of concepts that could be relevant in the diagnosis. Concepts are chosen so that specific questions can be addressed, e.g.: *Do the nuclei appear larger than usual? Do they present a vesicular texture with high chromatism?* To answer these questions about the nuclei area and texture, representative of the NGH nuclear pleomorphism, the segmentation

of the nuclei instances in the image is obtained by either manual annotations [14] or automatic segmentation [27]. We express the nuclei area as the sum of pixels in the nuclei contours, whereas the nuclei texture is described by Haralick's descriptors such as Angular Second Moment (ASM), contrast and correlation [18]. A summary of the concepts extracted and how to compute them is presented in Figure 6.

The performance of the RCV is evaluated by the determination coefficient of the regression $R2$, expressing the percentage of variation that is captured by the regression. This is used to check if the network is learning the concepts and in which network layers [14, 17]. Sensitivity scores are computed on testing patches from Camelyon17 as in [14]. The global relevance of the concept is estimated by either TCAV or Br scores.

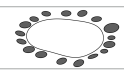
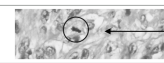
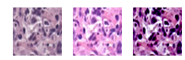
Concept	Clinical reference	Description	Visual examples	Magnification	Source	Type
Count of cavities	NGH tubular formation	tumour cells in gland structure	 	low	annotation or automated	D
Nuclei area	NGH nuclear pleomorphism	abnormality in size	 	high	annotation or automated	C
Nuclei Texture		vesicular appearance				
Mitotic count	NGH mitotic count	number of mitosis		high	annotation or automated	D
Nuclei density	Ki-67 protein expression	cell proliferation	 	any	annotation or automated	D
Staining	Staining procedure	dye applied on the tissues		any	metadata	D

Figure 6. Concept list derived for the breast histopathology application with information about the magnification level at which to extract them and whether the measures require a continuous (C) or discrete (D) representation

4 EVALUATION

4.1 Data Preprocessing

We report in Table 4 the evaluation of the execution times for the data preprocessing and patch extraction workflow. The two sampling processes, random and dense, are compared. The measurements were computed on the AGH site in Krakow, Poland.

The layer scalability to increasingly larger datasets and patient cohorts are shown in Figure 7. Scaling is possible by increasing the number of available nodes, for instance, from 100 to 1 000. In this case, approximately 50 000 patches can be extracted in less than 5 minutes with random sampling strategy, showing lin-

# WSIs	# Nodes	Sampling	Parallelization Type	CPU Time [s]
1	1	random	none	292
5	5	random	1 CPU/WSI	260
1	1	dense	none	18 400
1	100	dense	1 000 CPUs/WSI	3 000
5	500	dense	5 000 CPUs/WSI	7 530
5	1 000	dense	10 000 CPUs/WSI	3 780

Table 4. Measurements of execution time vs data sizes for extracting high resolution patches from the Camleyon17 dataset at the PROCESS AGH site. The WSI size is $100\,000 \times 100\,000$ pixels.

ear speed-up capability. Through the dense sampling strategy and by scaling the patch extraction to 1 000 nodes, we extracted 43 million patches, for a total of 7 TB of intermediate data. With 8 thousand nodes available for the computation, this would take approximately 1 hour with the SLURM parallelization technique.

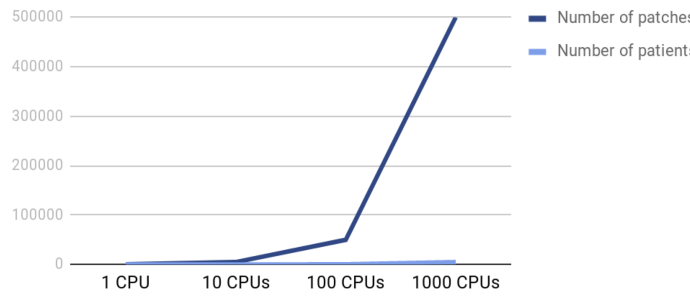


Figure 7. Layer 1 scalability to larger datasets and patient cohorts vs number of available nodes

4.2 Data Transfer Between HPC Sites

In addition to the computational time, we evaluate the time for data transfer between HPC centers, to establish whether this could constitute a possible bottleneck that would prevent the execution of two different software layers in two centers, e.g. Layer I at AGH and Layer II at UVA.

We show the cross-site staging results for transferring 30 GB of the Camelyon 16 dataset (3 % of the full dataset size, approximatively) in Table 5. Where available, i.e. for LRZ and UVA, we compare the Data Transfer Nodes connections. DTN nodes speed up trasfers of nearly 30 % compared to the standard SCP protocol.

	LRZ DTN	UVA DTN	LISA	AGH
LRZ DTN	–	405.32	25.53	32.17
UVA DTN	494.51	–	324.17	48.60
LISA	324.97	549.62	–	30.27
AGH	14.71	51.07	30.27	–

Table 5. Cross-site data staging speed for transfering 3 % of the Camelyon data with the gridFTP protocol. Measures are reported in Mb/s.

Figure 8 compares using standard SPC protocols for data transfer between DTN nodes against the dynamic reprogramming of DTNs with the FDT protocol through containers. For files smaller than 2 Gb, the overhead of deploying the containers on the fly is greater than the transfer time. For larger files, however, the overhead is amortized by the better performing FDT protocol.

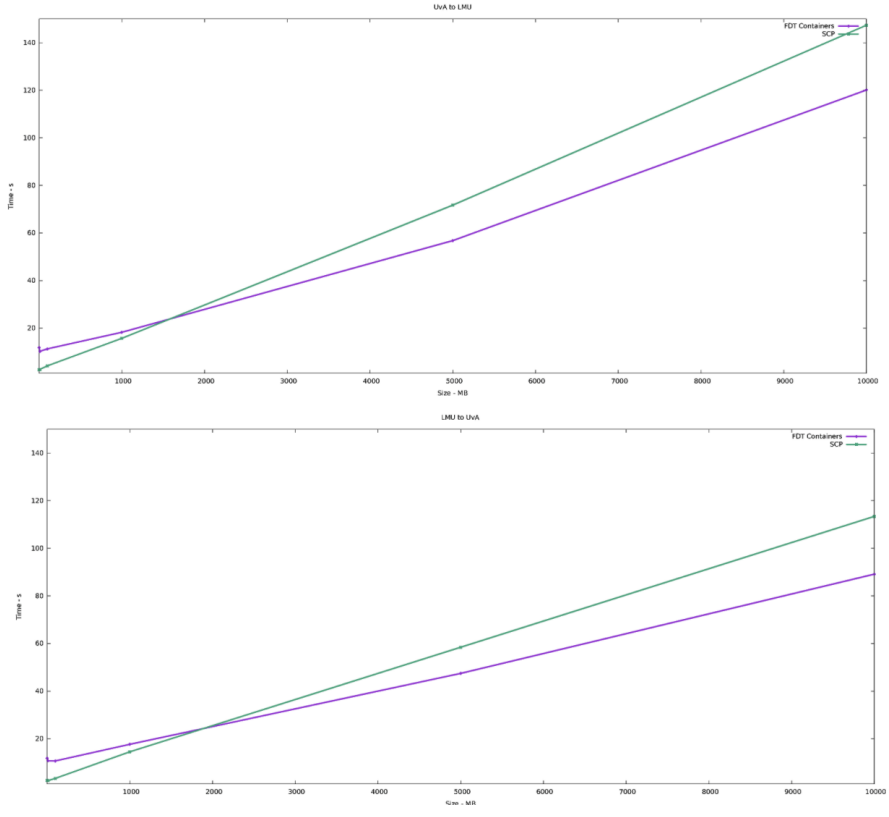


Figure 8. Comparison of SCP protocols vs FTD containerized approach for data transfer

4.3 Model Training

We compare in Figure 9 the training times (over 10 epochs) of two different architectures, namely ResNet50 and InceptionV3, on 50 Gb of training data. Less performant GPUs require a longer time to perform the training operations, with NVIDIA K80 requiring more than 7 hours to train the 26 million of parameters of ResNet50. This time reduces to slightly more than 1 hour when using the latest NVIDIA V100. The model parallelization on two GPUs, particularly on 2 NVIDIA V100 shows the scalability of the network training over multiple GPUs, requiring 1 hour and a half to train the 24 million of parameters of Inception V3.

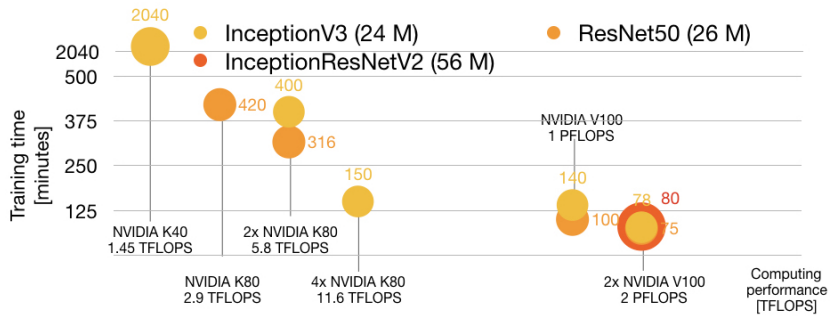


Figure 9. Comparison of ResNet and Inception average training times (on 50 Gb of training data) with single and distributed training on different GPUs. The number of parameters being trained is reported in brackets (M = millions). The floating point operations per second (FLOPS) are reported for each GPU configuration. The size of the circle is proportional to the number of parameters in each network.

The scalability of network training over larger datasets is compared for ResNet50 distributed on 2 NVIDIA V100 in Figure 10. This is insightful about the scalability of the combination of Layer I and Layer II estimating the training time per epoch for increasing dataset sizes.

4.4 Visualizations and Interpretability

Figure 11 shows some output heatmaps overlayed to the original input WSIs and compared to the manual tumor segmentations provided by the pathologist. The inference of nearly 10 thousand patches is distributed over 5 processes on a single NVIDIA V100, requiring less than 4 minutes to compute the heatmap for an input WSI (230 s). The network output is interpreted at the patch-level using gradCAM, gradCAM++ and LIME. Some examples are shown in Figure 11.

The concepts representative of the NGH nuclear pleomorphism are learned in early layers of the network, as shown in [14, 17]. Particularly, from the analysis of

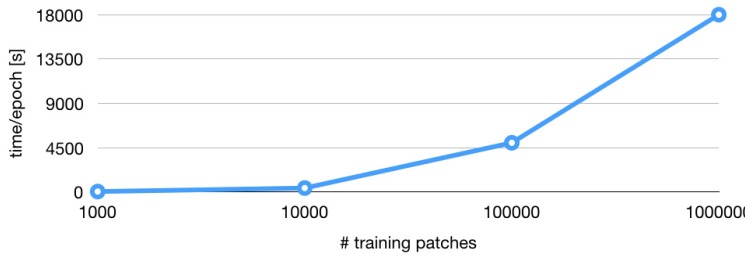


Figure 10. Training time per epoch vs increasingly larger data sizes for ResNet50 on $2 \times$ NVIDIA V100

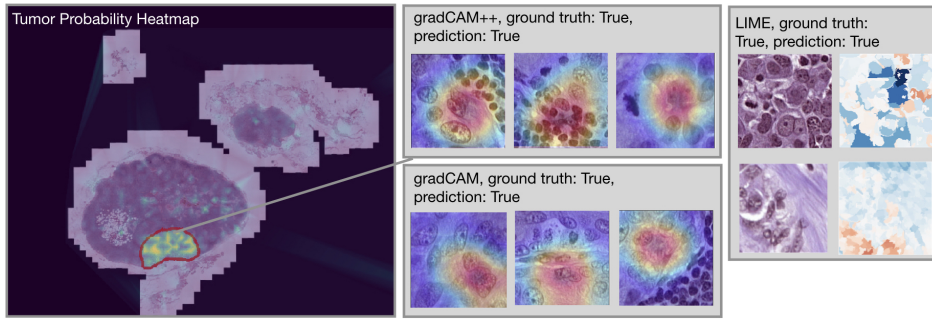


Figure 11. Visualization outputs. The heatmaps of tumor probability are computed by distributed inference over 5 models replicas on a single NVIDIA V100. The interpretability visualizations (on the right) were also computed on the same GPU.

the TCAV and Br scores it emerges that the contrast of the nuclei texture is relevant to the classification, with TCAV = 0.72 out of 1 and Br = 0.27.

5 DISCUSSION

The best performance for the data preprocessing and patch extraction pipeline (layer 1) is obtained when this layer is run on the HPC site. The scaling up of the dataset sizes is possible under the condition of a sufficiently large number of CPU cores on the computational site, which narrows down the computational time required by each operation. The results in Table 4 show the large benefits of optimizing the data extraction process on the HPC resources. In less than one hour, nearly 1 TB of data were extracted from 5 WSIs, with a computational requirement of one thousand nodes (with 10 CPU cores per node). Scaling this up led us to the extraction of 34 million of patches for a total of 7 TB that can be used for network training. Under the hypothesis of a large number of CPUs available, i.e. one per WSI, the computational requirements will not be a limit for data preprocessing, as the scalability requirements are almost linear.

Similarly, layer 2 was tested on different GPU servers, comparing performances of state-of-the-art networks and GPU types. The containerized approach allows users to deploy each layer on multiple sites without requiring specific expertise on the deployment site, thus leaving open different possibilities for deployment. The training times are consistently narrowed down by the distribution of training with model parallelism on multiple GPUs, as shown in Figures 9 and 10. The results on the distribution of network training show that the TFLOP of the GPUs is not a major limitation for scaling to larger datasets, with the performance of 4 NVIDIA K80 being close to that of a single NVIDIA V100 (150 minutes against 140 minutes respectively, as shown in Figure 9). Increasing data sizes leads to longer training times per epoch, as expected. In this case, even more TFLOPS are needed to scale up to millions of images. The training on 1 million images, however, can be performed in approximately 2 days with parallelization on 2 NVIDIA V100. The heatmap generation and interpretability analyses provide a direct visualization of the regions with a high probability of being tumorous. GradCAM, gradCAM++ and LIME provide various insights about the input regions responsible for the decision. RCVs further showed the importance of nuclei texture in the classification, with nuclei texture contrast being particularly relevant to the classification of patches of breast tissue. This is in accordance with the NHG grading system, which identifies hyperchromatism as a signal of nuclear atypia. It seems therefore that nuclear pleomorphism is taken into consideration during network training.

The results for each layer suggest that an optimal configuration would make the best use of the CPU clusters for data preprocessing, while network training should be performed on GPU servers. By testing data transfer times (see Table 5) we showed that the FTD containerized approach with DTNs would reduce the data staging bottleneck to the minimum.

6 CONCLUSION

We proposed a modular application that adapts with large flexibility to the different requirements of research in deep learning for histopathology and is deployable on HPC computing. The three layers can be deployed independently on the appropriate computing site to run different parts of the pipeline. The modularity of the proposed application embraces the foreseen future of digital pathology, being easy to deploy and offering large customization in terms of network parameters and choice of the training data. The parallelization of the different workflows improves the performance in terms of computational time. This is in line with the requirements of digital pathology, that, with increasingly larger datasets being collected and thus increasingly demanding tasks being set, is becoming demanding in terms of computational and storage requirements.

Moreover, the network training could be deployed independently on a private computational site, allowing users to fine-tune the network weights on sensitive data that cannot be shared.

Acknowledgements

This work is supported by the "PROviding Computing solutions for ExaScale ChallengeS" (PROCESS) project that has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 777533 and by the project APVV-17-0619 (U-COMP) "Urgent Computing for Exascale Data".

REFERENCES

[1] Aeffner, F.—Zarella, M. D.—Buchbinder, N.—Bui, M. M.—Goodman, M. R.—Hartman, D. J.—Lujan, G. M.—Molani, M. A.—Parwani, A. V.—Lillard, K.—Turner, O. C.—Vemuri, V. N. P.—Yuil-Valdes, A. G.—Bowman, D.: Introduction to Digital Image Analysis in Whole-Slide Imaging: A White Paper from the Digital Pathology Association. Journal of Pathology Informatics, Vol. 10, 2019, No. 9, doi: 10.4103/jpi.jpi_82_18.

[2] Alvarez-Melis, D.—Jaakkola, T. S.: Towards Robust Interpretability with Self-Explaining Neural Networks. Proceedings of the 32nd International Conference on Neural Information Processing Systems (NIPS 2018), 2018, pp. 7786–7795.

[3] Bonert, M.—Tate, A. J.: Mitotic Counts in Breast Cancer Should Be Standardized with a Uniform Sample Area. BioMedical Engineering OnLine, Vol. 16, 2017, No. 1, Art. No. 28, doi: 10.1186/s12938-016-0301-z.

[4] Bándi, P.—Geessink, O.—Manson, Q.—Van Dijk, M.—Balkenhol, M.—Hermsen, M.—Ehteshami Bejnordi, B.—Lee, B.—Paeng, K.—Zhong, A.—Li, Q.—Zanjani, F. G.—Zinger, S.—Fukuta, K.—Komura, D.—Ovtcharov, V.—Cheng, S.—Zeng, S.—Thagaard, J.—Dahl, A. B.—Lin, H.—Chen, H.—Jacobsson, L.—Hedlund, M.—Çetin, M.—Halici, E.—Jackson, H.—Chen, R.—Both, F.—Franke, J.—Küsters-Vandevelde, H.—Vreuls, W.—Bult, P.—van Ginneken, B.—van der Laak, J.—Litjens, G.: From Detection of Individual Metastases to Classification of Lymph Node Status at the Patient Level: The CAMELYON17 Challenge. IEEE Transactions on Medical Imaging, Vol. 38, 2019, No. 2, pp. 550–560, doi: 10.1109/TMI.2018.2867350.

[5] Campbell, C.—Mecca, N.—Duong, T.—Obeid, I.—Picone, J.: Expanding an HPC Cluster to Support the Computational Demands of Digital Pathology. 2018 IEEE Signal Processing in Medicine and Biology Symposium (SPMB), 2018, doi: 10.1109/SPMB.2018.8615614.

[6] Caruana, R.—Lou, Y.—Gehrke, J.—Koch, P.—Sturm, M.—Elhadad, N.: Intelligible Models for HealthCare: Predicting Pneumonia Risk and Hospital 30-Day Readmission. Proceedings of the 21th ACM SIGKDD International Conference on

Knowledge Discovery and Data Mining (KDD '15), ACM, 2015, pp. 1721–1730, doi: 10.1145/2783258.2788613.

[7] Chagpar, A.—Middleton, L. P.—Sahin, A. A.—Meric-Bernstam, F.—Kuerer, H. M.—Feig, B. W.—Ross, M. I.—Ames, F. C.—Singletary, S. E.—Buchholz, T. A.—Valero, V.—Hunt, K. K.: Clinical Outcome of Patients with Lymph Node-Negative Breast Carcinoma Who Have Sentinel Lymph Node Micrometastases Detected by Immunohistochemistry. Cancer, Vol. 103, 2005, No. 8, pp. 1581–1586, doi: 10.1002/cncr.20934.

[8] Cho, K.—Courville, A.—Bengio, Y.: Describing Multimedia Content Using Attention-Based Encoder-Decoder Networks. IEEE Transactions on Multimedia, Vol. 17, 2015, No. 11, pp. 1875–1886, doi: 10.1109/TMM.2015.2477044.

[9] Dean, J.—Corrado, G.—Monga, R.—Chen, K.—Devin, M.—Mao, M.—Ranzato, M.—Senior, A.—Tucker, P.—Yang, K.—Le, Q.—Ng, A.: Large Scale Distributed Deep Networks. In: Pereira, F., Burges, C. J. C., Bottou, L. Weinberger, K. Q. (Eds.): Advances in Neural Information Processing Systems 25 (NIPS 2012), 2012, pp. 1223–1231.

[10] Fong, R. C.—Vedaldi, A.: Interpretable Explanations of Black Boxes by Meaningful Perturbation. Proceedings of the 2017 IEEE International Conference on Computer Vision, Venice, Italy, 2017, pp. 3449–3457, doi: 10.1109/ICCV.2017.371.

[11] Freitas, A. A.: Comprehensible Classification Models: A Position Paper. ACM SIGKDD Explorations Newsletter, Vol. 15, 2014, No. 1, pp. 1–10, doi: 10.1145/2594473.2594475.

[12] Giuliano, A. E.—Ballman, K. V.—McCall, L.—Beitsch, P. D.—Brennan, M. B.—Kelemen, P. R.—Ollila, D. W.—Hansen, N. M.—Whitworth, P. W.—Blumencranz, P. W.—Leitch, A. M.—Saha, S.—Hunt, K. K.—Morrow, M.: Effect of Axillary Dissection vs. No Axillary Dissection on 10-Year Overall Survival Among Women with Invasive Breast Cancer and Sentinel Node Metastasis: The ACOSOG Z0011 (Alliance) Randomized Clinical Trial. JAMA, Vol. 318, 2017, No. 10, pp. 918–926, doi: 10.1001/jama.2017.11470.

[13] Goyal, P.—Dollár, P.—Girshick, R.—Noordhuis, P.—Wesolowski, L.—Kyrola, A.—Tulloch, A.—Jia, Y.—He, K.: Accurate, Large Minibatch SGD: Training ImageNet in 1 Hour. 2017, arXiv preprint arXiv:1706.02677.

[14] Graziani, M.—Andrearczyk, V.—Müller, H.: Regression Concept Vectors for Bidirectional Explanations in Histopathology. In: Stoyanov, D. et al. (Eds.): Understanding and Interpreting Machine Learning in Medical Image Computing Applications (MLCN 2018, DLF 2018, IMIMIC 2018). Springer, Cham, Lecture Notes in Computer Science, Vol. 11038, 2018, pp. 124–132, doi: 10.1007/978-3-030-02628-8_14.

[15] Graziani, M.—Andrearczyk, V.—Müller, H.: Visualizing and Interpreting Feature Reuse of Pretained CNNs for Histopathology. Irish Machine Vision and Image Processing Conference (IMVIP 2019), Dublin, Ireland, 2019.

[16] Graziani, M.—Lompech, T.—Müller, H.—Depeursinge, A.—Andrearczyk, V.: Interpretable CNN Pruning for Preserving Scale-Covariant Features in Medical Imaging. In: Cardoso, J. et al. (Eds.): Interpretable and Annotation-Efficient Learning for Medical Image Computing (IMIMIC 2020, MIL3ID

2020, LABELS 2020). Springer, Cham, in cooperation with MICCAI, Lecture Notes in Computer Science, Vol. 12446, 2020, pp. 23–32, doi: 10.1007/978-3-030-61166-8_3.

[17] Graziani, M.—Andrearczyk, V.—Marchand-Maillet, S.—Müller, H.: Concept Attribution: Explaining CNN Decisions to Physicians. Computers in Biology and Medicine, Vol. 123, 2020, Art. No. 103865, doi: 10.1016/j.compbiomed.2020.103865.

[18] Haralick, R. M.—Shanmugam, K.—Dinstein, I.: Textural Features for Image Classification. IEEE Transactions on Systems, Man, and Cybernetics, Vol. SMC-3, 1973, No. 6, pp. 610–621, doi: 10.1109/TSMC.1973.4309314.

[19] Hayakawa, T.—Prasath, V. B. S.—Kawanaka, H.—Aronow, B. J.—Tsuruoka, S.: Computational Nuclei Segmentation Methods in Digital Pathology: A Survey. Archives of Computational Methods in Engineering, 2019, pp. 1–13, doi: 10.1007/s11831-019-09366-4.

[20] Kim, B.—Shah, J. A.—Doshi-Velez, F.: Mind the Gap: A Generative Approach to Interpretable Feature Selection and Extraction. In: Cortes, C., Lawrence, N., Lee, D., Sugiyama, M., Garnett, R. (Eds.): Advances in Neural Information Processing Systems 28 (NIPS 2015), 2015, pp. 2260–2268.

[21] Kurth, T.—Zhang, J.—Satish, N.—Racah, E.—Mitliagkas, I.—Patwary, M. M. A.—Malas, T.—Sundaram, N.—Bhimji, W.—Smorkalov, M.—Deslippe, J.—Shiryaev, M.—Sridharan, S.—Prabhat—Dubey, P.: Deep Learning at 15PF: Supervised and Semi-Supervised Classification for Scientific Data. Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis (SC '17), ACM, 2017, Art. No. 7, 11 pp., doi: 10.1145/3126908.3126916.

[22] Li, J.—Yang, S.—Huang, X.—Da, Q.—Yang, X.—Hu, Z.—Duan, Q.—Wang, C.—Li, H.: Signet Ring Cell Detection with a Semi-Supervised Learning Framework. In: Chung, A., Gee, J., Yushkevich, P., Bao, S. (Eds.): Information Processing in Medical Imaging (IPMI 2019). Springer, Cham, Lecture Notes in Computer Science, Vol. 11492, 2019, pp. 842–854, doi: 10.1007/978-3-030-20351-1_66.

[23] Lin, Y.—Han, S.—Mao, H.—Wang, Y.—Dally, W. J.: Deep Gradient Compression: Reducing the Communication Bandwidth for Distributed Training. 2017, arXiv preprint arXiv:1712.01887.

[24] Lipton, Z. C.: The Mythos of Model Interpretability. Communication of ACM, Vol. 61, 2018, No. 10, pp. 36–43, doi: 10.1145/3233231.

[25] Lu, C.—Romo-Bucheli, D.—Wang, X.—Janowczyk, A.—Ganesan, S.—Gilmore, H.—Rimm, D.—Madabhushi, A.: Nuclear Shape and Orientation Features from H & E Images Predict Survival in Early-Stage Estrogen Receptor-Positive Breast Cancers. Laboratory Investigation, Vol. 98, 2018, No. 11, pp. 1438–1448, doi: 10.1038/s41374-018-0095-7.

[26] Madabhushi, A.—Lee, G.: Image Analysis and Machine Learning in Digital Pathology: Challenges and Opportunities. Medical Image Analysis, Vol. 33, 2016, pp. 170–175, doi: 10.1016/j.media.2016.06.037.

[27] Otálora, S.—Atzori, M.—Khan, A.—Jimenez-del-Toro, O.—Andrearczyk, V.—Müller, H.: Systematic Comparison of Deep Learning Strategies for Weakly Supervised Gleason Grading. Medical Imaging 2020: Digital Pathology. Proceedings of the SPIE, Vol. 11320, 2020, Art. No. 113200L, doi: 10.1117/12.2548571.

[28] Otálora, S.—Marini, N.—Müller, H.—Atzori, M.: Semi-Weakly Supervised Learning for Prostate Cancer Image Classification with Teacher-Student Deep Convolutional Networks. In: Cardoso, J. et al. (Eds.): Interpretable and Annotation-Efficient Learning for Medical Image Computing (IMIMIC 2020, MIL3ID 2020, LABELS 2020). Springer, Cham, Lecture Notes in Computer Science, Vol. 12446, 2020, pp. 193–203, doi: 10.1007/978-3-030-61166-8_21.

[29] Rabe, K.—Snir, O. L.—Bossuyt, V.—Harigopal, M.—Celli, R.—Reisenbichler, E. S.: Interobserver Variability in Breast Carcinoma Grading Results in Prognostic Stage Differences. Human Pathology, Vol. 94, 2019, pp. 51–57, doi: 10.1016/j.humpath.2019.09.006.

[30] Rakha, E. A.—El-Sayed, M. E.—Lee, A. H. S.—Elston, C. W.—Grainge, M. J.—Hodi, Z.—Blamey, R. W.—Ellis, I. O.: Prognostic Significance of Nottingham Histologic Grade in Invasive Breast Carcinoma. Journal of Clinical Oncology, Vol. 26, 2008, No. 19, pp. 3153–3158, doi: 10.1200/JCO.2007.15.5986.

[31] Ribeiro, M. T.—Singh, S.—Guestrin, C.: Why Should I Trust You?: Explaining the Predictions of Any Classifier. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '16), ACM, 2016, pp. 1135–1144, doi: 10.1145/2939672.2939778.

[32] Schnitt, S. J.—Connolly, J. L.—Tavassoli, F. A.—Fechner, R. E.—Kempson, R. L.—Gelman, R.—Page, D. L.: Interobserver Reproducibility in the Diagnosis of Ductal Proliferative Breast Lesions Using Standardized Criteria. The American Journal of Surgical Pathology, Vol. 16, 1992, No. 12, pp. 1133–1143, doi: 10.1097/00000478-199212000-00001.

[33] Selvaraju, R. R.—Cogswell, M.—Das, A.—Vedantam, R.—Parikh, D.—Batra, D.: Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization. 2017 IEEE International Conference on Computer Vision (ICCV), 2017, pp. 618–626, doi: 10.1109/ICCV.2017.74.

[34] Shen, S.—Han, S. X.—Aberle, D. R.—Bui, A. A.—Hsu, W.: An Interpretable Deep Hierarchical Semantic Convolutional Neural Network for Lung Nodule Malignancy Classification. Expert Systems with Applications, Vol. 128, 2019, pp. 84–95, doi: 10.1016/j.eswa.2019.01.048.

[35] Siegel, R. L.—Miller, K. D.—Jemal, A.: Cancer Statistics, 2019. CA: A Cancer Journal for Clinicians, Vol. 69, 2019, No. 1, pp. 7–34, doi: 10.3322/caac.21551.

[36] Simonyan, K.—Vedaldi, A.—Zisserman, A.: Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps. Computing Research Repository (CoRR), 2013, arXiv:1312.6034, http://arxiv.org/abs/1312.6034.

[37] Sundararajan, M.—Taly, A.—Yan, Q.: Axiomatic Attribution for Deep Networks. Proceedings of the 34th International Conference on Machine Learning (ICML '17), JMLR.org, Proceedings of Machine Learning Research (PMLR), Vol. 70, 2017, pp. 3319–3328.

[38] van Diest, P. J.—van Deurzen, C. H. M.—Cserni, G.: Pathology Issues Related to SN Procedures and Increased Detection of Micrometastases and Isolated Tumor Cells. Breast Disease, Vol. 31, 2010, No. 2, pp. 65–81, doi: 10.3233/BD-2010-0298.

[39] Wang, F.—Aji, A.—Liu, Q.—Saltz, J. H.: Hadoop-GIS: A High Performance Query System for Analytical Medical Imaging with Mapreduce. Technical Report CCI-TR-2001-3, Emory University, Atlanta, USA, 2011, pp. 1–13.

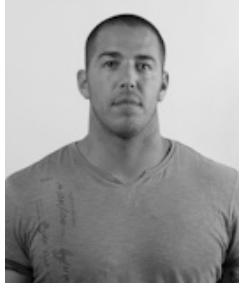
[40] Wei, X. S.—Wu, J.—Zhou, Z. H.: Scalable Multi-Instance Learning. 2014 IEEE International Conference on Data Mining (ICDM), Shenzhen, China, 2014, pp. 1037–1042, doi: 10.1109/ICDM.2014.16.

[41] Xu, Y.—Li, Y.—Shen, Z.—Wu, Z.—Gao, T.—Fan, Y.—Lai, M.—Chang, E. I-C.: Parallel Multiple Instance Learning for Extremely Large Histopathology Image Analysis. BMC Bioinformatics, Vol. 18, 2017, Art. No. 360, 15 pp., doi: 10.1186/s12859-017-1768-8.

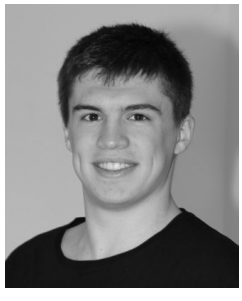
[42] Zanjani, F. G.—Zinger, S.—De, P. N.: Automated Detection and Classification of Cancer Metastases in Whole-Slide Histopathology Images Using Deep Learning. 2017.

[43] Zeiler, M. D.—Fergus, R.: Visualizing and Understanding Convolutional Networks. Computing Research Repository (CoRR), 2013, arXiv:1311.2901, `http://arxiv.org/abs/1311.2901`.

Mara Graziani is a third-year Ph.D. student with double affiliation at the University of Geneva and at the University of Applied Sciences of Western Switzerland. With her research, she aims at improving the interpretability of machine learning systems for healthcare by a human-centric approach. She was a visiting student at the Martinos Center, part of Harvard Medical School in Boston, MA, USA to analyze the interaction between clinicians and deep learning systems. From her background of IT engineering, she was awarded the Engineering Department Award for completing the M.Phil. in machine learning, speech and language at the University of Cambridge, UK in 2017.



Ivan Eggel is Senior Research Associate of the University of Applied Sciences of Western Switzerland. His research interest focuses on developing applications for information retrieval in healthcare and in cloud computing. He participated in the organization of several challenges such as ImageCLEF and VISCERAL.



François Deligand is a Bachelor's student in mathematics and informatics at INP-ENSEEIHT in Toulouse, France. He contributed to the experiments during his internship at HES-SO Valais.

Vincent Andrearczyk is currently Senior Researcher at the University of Applied Sciences and Arts Western Switzerland with a research focus on deep learning for medical image analysis and texture feature extraction. He received a double Masters degree in electronics and signal processing from ENSEEIHT, France and Dublin City University in 2012 and 2013, respectively. He completed his Ph.D. degree on deep learning for texture and dynamic texture analysis at Dublin City University in 2017.

Martin Bobak is Scientist at the Institute of Informatics (Slovak Academy of Sciences, Bratislava, Slovakia), in the Department of Parallel and Distributed Information Processing. He started working at the institute in 2013, defended his dissertation thesis at the institute in 2017, became Member of the Scientific Board of the institute, and Guest Handling Editor in the CC journal Computing and Informatics. His field of research is cloud computing and the architectures of distributed cloud-based applications. He is the author of numerous scientific publications and has participated in several European and Slovak R & D projects.

Henning Müller is Full Professor at the HES-SO Valais and responsible for the eHealth unit of the school. He is also Professor at the Medical Faculty of the University of Geneva and has been on sabbatical at the Martinos Center, part of Harvard Medical School in Boston, MA, USA to focus on research activities. He is the coordinator of the ExaMode EU project, was coordinator of the Khresmoi EU project, scientific coordinator of the VISCERAL EU project and is the initiator of the ImageCLEF benchmark that has run medical tasks since 2004. He has authored over 600 scientific papers with more than 17 000 citations and is in the editorial board of several journals.