Computer Methods and Programs in Biomedicine 239 (2023) 107631

Contents lists available at ScienceDirect

Computer Methods and Programs in Biomedicine

journal homepage: www.elsevier.com/locate/cmpb

ChampKit: A framework for rapid evaluation of deep neural networks for patch-based histopathology classification

Jakub R. Kaczmarzyk[a,c,*], Rajarsi Gupta[a], Tahsin M. Kurc[a], Shahira Abousamra[b], Joel H. Saltz[a,1], Peter K. Koo[c,1]

[a] Department of Biomedical Informatics, Stony Brook Medicine, 101 Nicolls Rd, Stony Brook, 11794, NY, USA
[b] Department of Computer Science, Stony Brook University, Stony Brook, NY, USA
[c] Simons Center for Quantitative Biology, 1 Bungtown Rd, Cold Spring Harbor, 11724, NY, USA

ARTICLE INFO

Article history:
Received 19 January 2023
Revised 23 April 2023
Accepted 28 May 2023

Keywords:
Computational pathology
Histopathology
Deep learning
Benchmarks
Classification

ABSTRACT

Background and Objective: Histopathology is the gold standard for diagnosis of many cancers. Recent advances in computer vision, specifically deep learning, have facilitated the analysis of histopathology images for many tasks, including the detection of immune cells and microsatellite instability. However, it remains difficult to identify optimal models and training configurations for different histopathology classification tasks due to the abundance of available architectures and the lack of systematic evaluations. Our objective in this work is to present a software tool that addresses this need and enables robust, systematic evaluation of neural network models for patch classification in histology in a light-weight, easy-to-use package for both algorithm developers and biomedical researchers.

Methods: Here we present *ChampKit* (Comprehensive Histopathology Assessment of Model Predictions toolKit): an extensible, fully reproducible evaluation toolkit that is a one-stop-shop to train and evaluate deep neural networks for patch classification. ChampKit curates a broad range of public datasets. It enables training and evaluation of models supported by `timm` directly from the command line, without the need for users to write any code. External models are enabled through a straightforward API and minimal coding. As a result, Champkit facilitates the evaluation of existing and new models and deep learning architectures on pathology datasets, making it more accessible to the broader scientific community. To demonstrate the utility of ChampKit, we establish baseline performance for a subset of possible models that could be employed with ChampKit, focusing on several popular deep learning models, namely ResNet18, ResNet50, and R26-ViT, a hybrid vision transformer. In addition, we compare each model trained either from random weight initialization or with transfer learning from ImageNet pretrained models. For ResNet18, we also consider transfer learning from a self-supervised pretrained model.

Results: The main result of this paper is the ChampKit software. Using ChampKit, we were able to systemically evaluate multiple neural networks across six datasets. We observed mixed results when evaluating the benefits of pretraining versus random intialization, with no clear benefit except in the low data regime, where transfer learning was found to be beneficial. Surprisingly, we found that transfer learning from self-supervised weights rarely improved performance, which is counter to other areas of computer vision.

Conclusions: Choosing the right model for a given digital pathology dataset is nontrivial. ChampKit provides a valuable tool to fill this gap by enabling the evaluation of hundreds of existing (or user-defined) deep learning models across a variety of pathology tasks. Source code and data for the tool are freely accessible at https://github.com/SBU-BMI/champkit.

* Corresponding author.
E-mail addresses: jakub.kaczmarzyk@stonybrookmedicine.edu (J.R. Kaczmarzyk), joel.saltz@stonybrookmedicine.edu (J.H. Saltz), koo@cshl.edu (P.K. Koo).
[1] Senior authors.

https://doi.org/10.1016/j.cmpb.2023.107631

1. Introduction

Histopathology is the gold standard for cancer diagnosis. Pathologists review tissue slides visually, often alternating among different magnifications and fields of view. However, examining the entirety of a slide at its highest resolution (e.g., 40x) and examining a large number of slides for a study are very challenging tasks, due to the sheer amount of time required. There is growing interest in computer-assisted analysis of histopathology images with deep learning. Physical pathology slides can be digitized at high resolution, in a process known as digital pathology, thanks to advanced tissue scanners. Digital pathology is at its core a big data problem, as these images are large (e.g., 100,000 by 80,000 pixels) and take up several gigabytes of storage.

There is significant interest in developing deep learning computer vision algorithms for digital pathology images in order to label digital pathology images at high resolutions [3–12]. Digital pathology images are too large for conventional deep learning algorithms and hardware, so the images are separated into patches for use with deep learning algorithms. There are several types of computer vision approaches for digital pathology, including slide-level classification, patch-based semantic segmentation, patch-based object detection, and patch-based classification. In this manuscript, we focus exclusively on patch-based classification tasks. Many studies in digital pathology have introduced patch-based deep learning algorithms that seek to characterize digital pathology images. Owing in large part to large-scale, publicly available, multi-cancer projects like The Cancer Genome Atlas, patch classification has been explored in a large number of cancers, including breast [13–15], lung [16–18], pancreas [19–23], and prostate [24,25] cancers to name a few. Other well-studied applications of digital pathology patch classification include detecting microsatellite instability [6,26–29], tumor cells [30–35], and tumor-infiltrating lymphocytes [36–42].

While deep learning can be a powerful tool, planning a deep learning study in digital pathology is nontrivial [43]. There are several challenges, including how to choose the best deep learning model for the task, how to choose hyperparameters of models, and how to curate training and evaluation data. The choice of models is a high-dimensional problem. There are hundreds of neural network architectures to choose from, and within each there are hyperparameters that can greatly affect classification performance [43]. Model repositories exist that facilitate the use of neural networks pretrained on common natural image datasets like ImageNet. One such repository is `timm` [1], which contains dozens of model architectures and hundreds of sets of pretrained weights. Other examples include the PyTorch and TensorFlow model hubs. However, there is an unmet need for digital pathology, because *these repositories do not contain pathology-specific models*, though models trained on pathology data using self-supervision have recently come online [44–46]. Tools also exist to run benchmarking experiments [47–49] but these do not curate data specific to digital pathology.

There are several sources of public digital pathology data, and *it would be greatly beneficial to the digital pathology community to curate these data in one location and prepare them for immediate use with deep learning workflows*. Although the pieces required to evaluate patch-based digital pathology models exist as part of separate studies, a user would have to perform a complex process to curate relevant data, prepare reference implementations of neural network architectures, download pretrained weights, and implement relevant performance metrics. *A better solution is a toolkit that streamlines these components for researchers* [43]. The resulting framework should also be reproducible and simple for users to run, to mitigate any potential difference in results from differences in how the code was run. Finally, while there has been progress in developing such a toolkit to evaluate slide-level classification [50], *there does not yet exist a reproducible, extensible toolkit coupled with a comprehensive set of benchmark datasets for patch-level analysis*.

To address these unmet needs in digital pathology, we introduce *ChampKit* (Comprehensive Histopathology Assessment of Model Predictions toolKit), an easy-to-use toolkit that focuses on the evaluation of neural network models for computational pathology image analysis (Fig. 1). The target users of ChampKit are (1) methods research groups interested in systematically and quickly evaluating their deep learning methods against a set of state-of-the-art (SOTA) methods with different pretraining and transfer learning configurations, and (2) biomedical research groups interested in finding and fine-tuning the best models to analyze a collection of whole slide images. Both research communities would benefit from the ability to quickly and systematically evaluate a set of SOTA (pre-trained) methods, and Champkit enables these two use cases in an easy-to-use package. In Section 4, we recount how our group has used ChampKit to identify optimal models for Gleason grade classification. Importantly, ChampKit complements existing tools. It makes use of the `timm` model repository and extends it with the addition of pathology-specific pretrained models. Transfer learning has shown inconsistent benefits in digital pathology [51–53], so one use of ChampKit could be to comprehensively characterize the impact of transfer learning across models and pathology datasets. ChampKit also curates multiple public datasets for patch-based classification. The toolkit and datasets adhere to FAIR principles [54,55], which enhances the reusability and reproducibility of this work. ChampKit is meant to simplify the training and evaluation of deep learning models for patch-based classification.

The main contribution of this paper is the ChampKit software. To demonstrate the utility of ChampKit, we perform a study that establishes baseline performance of several existing models across six diverse classification tasks across various cancers through publicly available datasets curated by the toolkit.

1.1. Related work

Benchmarking of deep neural networks is a well-studied field, and multiple solutions exist in this space. BIAFLOWS is a benchmarking and deployment platform for microscopy image workflows. It supports many problem types, including object segmentation and particle tracking [56], and it implements metrics specific to each supported problem type. OpenML is another platform that allows for the creation and sharing of machine learning benchmarks and emphasizes community contributions of benchmark results [57]. Ludwig is a benchmarking toolkit that enables configurable, personalized benchmarking [49]. ShinyLearner is a tool to benchmark classification of tabular data [58]. Weights and Biases [2] provides many methods for hyperparameter search, experiment logging, and experiment visualization, all of which are useful for benchmarking. The Python package `timm` [1] allows easy access to hundreds of deep learning models for image classification, many of which are pretrained on ImageNet [59], and thus can facilitate analysis across many models. To our knowledge, however, there are few benchmarking solutions designed specifically for histopathology. The work of Laleh et al. [50], for example, provides a benchmark for weakly-supervised, specimen-level classification in digital pathology. These related projects have inspired the development of ChampKit as a benchmarking toolkit specific to patch-based classification in digital pathology.

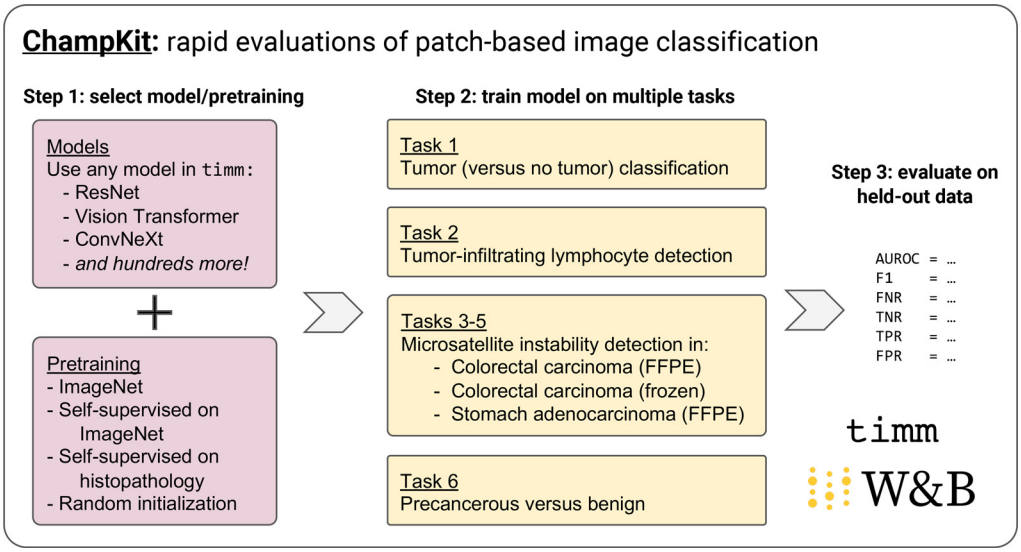


Fig. 1. Overview of ChampKit. ChampKit enables systematic comparisons of model architectures and transfer learning across several patch-based image classification datasets. First, users select a model and pretrained weights from those available in timm [1] or a custom model with specfication of pretrained weights or random initialization. Second, the models are trained on multiple tasks using identical training hyperparameters. Third, the trained models are evaluated on held-out test data for each task. Performance is tracked with Weights and Biases [2].

2. Materials and methods

2.1. ChampKit

ChampKit is a one-stop-shop for systematic exploration of deep learning architectures, training strategies, and transfer learning across patch-level histopathology classification tasks. Research in deep learning architectures moves quickly, so ChampKit integrates the `timm` [1] model repository to provide access to hundreds of (pretrained) deep learning models, enabling evaluation across different transfer learning schemes from ImageNet [59] or from self-supervision on histopathology images [46,60] or training from scratch. ChampKit enables rapid evaluation of hyperparameters using Weights & Biases [2] and neural network architectures through `timm` [1] and `torchvision` [61]. Users can incorporate their own datasets and models as well. Importantly, ChampKit does not require any coding — users can use configuration files and command line programs to run training and evaluation. Detailed instructions to use and extend ChampKit are available in the GitHub repository.[1]

Toolkit There are two main use cases for ChampKit: 1) to identify an optimal classification model for one's own patch classification dataset, and 2) to benchmark models across multiple datasets, including those that are downloaded with ChampKit. For both use cases, the toolkit includes scripts to perform an end-to-end analysis: prepare datasets, download pretrained models, train models, and evaluate them on held-out test data. Weights and Biases [2] is used to log experiment parameters, visualize results across multiple models and training configurations, and orchestrate hyperparameter searches. Three hyperparameter search strategies are available currently via Weights and Biases, and from most to least efficient they are Bayesian, random, and grid search [62,63]. Users can design their own hyperparameter searches to find the optimal model for their own dataset, and the search configurations included in our code repository can be used as a starting point. In the following sections, we discuss how one can identify optimal models, use their own datasets, and use other models with ChampKit.

Identifying optimal models ChampKit includes a Python script to evaluate trained models on unseen data. This script calculates various performance metrics, including area under the receiver operating characteristic curve (AUROC), accuracy, F1 score, and a confusion matrix, and it also saves plots of these values across models. These evaluations support binary and multi-class classification tasks and will show the performance results per class. This is especially helpful in unbalanced datasets, so that one can determine whether the trained models perform well on the underrepresented class(es). The evaluation script then identifies the best model per evaluation metric and prints these results. The evaluation scores for each model are saved to a spreadsheet so that one can perform further processing if desired. Using this tool in ChampKit, one can choose the optimal model for one's patch classification dataset.

Datasets ChampKit has currently curated six patch-based image classification tasks for: (1) tumor (versus no tumor) classification, (2) tumor-infiltrating lymphocyte detection, (3–5) microsatellite instability detection across different cancers and/or preparations, and (6) precancerous versus benign classification. These datasets represent a wide variety of tissue types, cancers, tasks, and sample sizes (see Table S1). This diversity represents a starting point for benchmarking patch classification in digital pathology and enables users to explore models that might generalize across these datasets or be ideal for certain data characteristics. In addition to serving as built-in datasets for exploration, these datasets serve to demonstrate the capabilities of ChampKit as a benchmarking toolkit. In Section 3, we describe the performance of several models on each dataset. ChampKit includes reproducible scripts to download all datasets, with the exception of the MHIST dataset [64], which requires completing an online form (an automated email is then sent with a download URL). The datasets for each task are curated from different studies [64–68] and were selected using the following guidelines: (1) designed for patch-based classification, (2) accessible without the need to make an online account, (3) dataset is versioned, (4) sufficient size (at least a few thousand images), (5) diversity of tasks, and (6) in our judgement, important to the biomedical community. Five of the six datasets are hosted

[1] https://github.com/SBU-BMI/champkit

on Zenodo, which stores static copies of the data. If these datasets are updated, the updated datasets can easily be incorporated into ChampKit. Each dataset was split into training, validation, and testing partition (see Table S1). Models were trained on the training set and the validation set was used to evaluate performance at the end of each epoch. Final models were chosen based on performance on the validation set. The test sets were used for final evaluation, and all results are reported on the test set. All data are downloaded when the user initializes the ChampKit repository, accessible without any requirements for registration.

Using a new dataset Beyond the six benchmark datasets, ChampKit allows for integration with custom datasets that are framed as binary or multi-class patch classification. Datasets must be organized in an ImageNet-like directory structure, in which the top-level directory contains directories `train`, `val`, and `test`, and each of those contains sub-directories of the different classes (e.g., `tumor-positive`, `tumor-negative`) with the corresponding patches. Images can be of any size and will be resized during training and evaluation – the size is configurable. Indeed, we demonstrate the creation and use of a new dataset in Section 3.5.

Preparing a dataset from annotated whole slide images ChampKit expects a dataset of patches in PNG or JPEG format. If one has a set of annotated whole slide images, where the annotations indicate regions in the slide that belong to a label (e.g., "normal epithelium"), then existing tools can be used to export patches from these labeled regions. These tools include QuPath [69], OpenSlide [70], and Large Image [71]. When exporting patches, users should consider the physical resolution of the patches and should use a consistent resolution for all patches. For example, a user may choose to extract patches of 224×224 pixels at 0.5 μm per pixel. Patches are also assumed to have mutually exclusive labels, so it is important to ensure that a patch is not a member of two annotated regions with different labels. Next, the extracted patches must be split into training, validation, and test sets (to avoid data leakage, patches from any single specimen or patient should not be a member of multiple subsets). As described in the preceding paragraph, all images that belong to a particular class should be placed in a class-specific directory. If, for example, the task is binary classification between "normal epithelium" and "neoplastic epithelium", the directories for the training set would be named `train/normal_epithelium/` and `train/neoplastic_epithelium/`, and the directories would contain all training images of normal epithelium and neoplastic epithelium. This would be repeated for the validation and test subsets. After the dataset is prepared in this fashion, one can use ChampKit to identify optimal classification models for the dataset. Section 3.5 describes the creation of a patch classification dataset from annotated whole slide images in the PANDA dataset of prostate biopsies [72].

Using custom models While ChampKit provides access to a wide range of deep learning architectures and pre-trained models via `timm` [1], it may still be desirable to tweak an existing model, employ a custom model, or use pre-trained weights from pathology-specific models. Image classification models specific to pathology are being published regularly, and a user may want to apply these models to their own datasets. In many cases, pathology-specific models use architectures found in `timm`, but the weights will be specific to a pathology dataset. In that case, one can provide the name of the network architecture, the image normalization parameters, and a path to the pre-trained weights (an example of this is provided in the ChampKit code repository). If the network is not one found in `timm`, one can include the PyTorch [73] implementation of the architecture as well as image normalization parameters. The ChampKit code repository includes an example of using pretrained weights from a pathology-specific model [60], and we report results using this model in this manuscript. The self-supervised model uses the ResNet18 architecture and was trained on a large histopathology dataset in a self-supervised fashion. ChampKit itself does not support pretraining of models, but it does leverage pretrained models from published literature. Future versions of ChampKit may include other pathology-specific models, and the authors welcome community contributions of these models, which can be made through a pull request to our GitHub repository.

Criteria for reproducibility We strive to achieve the silver standard of reproducibility as defined by [74]. The silver standard has three criteria:

1. Software dependencies are all prepared with a single command,
2. Documentation of how to run scripts and in which order,
3. Random elements must be made deterministic.

ChampKit satisfies all three requirements. All dependencies are installed using the `conda` Python environment manager, and datasets are downloaded with a single command. The README of the source code includes extensive documentation of how to run the training and evaluation scripts, and the random number generators used during training are seeded, so that with the same seed, results will be identical or almost identical. Dataset preparation is also designed to be reproducible, meaning that when one initializes ChampKit, one is guaranteed to have exactly the same copy of the data as was reported in this manuscript. This is accomplished in two ways: 1) five of the six datasets in ChampKit are hosted on Zenodo, which preserves immutable versions of data; and 2) the MD5 hashes of the downloaded data are validated at download time to ensure that the user has the intended version of the dataset. In some cases, the downloaded dataset is minimally curated, including moving image patches into an appropriate directory structure and splitting the dataset into training, validation, and test splits. These splits are done with seeded pseudorandom number generators so that the results are deterministic. Data download and curation is all done automatically when one initializes the ChampKit repository with the `setup.sh` script.

2.2. Experiments

The main purpose of this work is to introduce ChampKit as a toolkit to rapidly evaluate deep neural networks on histopathology patch classification tasks. To demonstrate the utility of ChampKit, we systematically trained several models commonly used in histopathology, namely ResNet18 and ResNet50 [75], to establish baseline performance for each benchmark task. We also include a hybrid vision transformer (R26-ViT) [76], which has shown promise at image classification in smaller data regimes. All models included comparisons of transfer learning from image classification on ImageNet-1K [59] and training from random weight initialization. ResNet18 comparisons also included transfer learning from a model trained using self-supervision on histopathology data [46,60]. (ChampKit is designed to benchmark existing architectures from existing weights, and thus it does not support pretraining of models.) All models were implemented in `timm` using PyTorch, and pretrained ImageNet weights were accessed via `timm`, while the weights from self-supervised pre-training for ResNet18 were downloaded from [60]. The AdamW optimizer was used with a learning rate of 0.001 [77], and models optimized cross entropy loss. The learning rate was warmed up from 1e-6 to 0.001 over the first 10 epochs, followed by cosine decay set for 500 epochs. Early stopping was enabled after 30 epochs based on validation cross entropy loss with a patience of 20 epochs. All models used a dropout rate of 0.3. Each experiment was run on a single NVIDIA Quadro RTX 8000 GPU with 48GB of video memory.

Data processing RGB channels were normalized with means (0.485, 0.456, 0.406) and standard deviations (0.229, 0.224, 0.225);

these values are the means and standard deviations of the ImageNet training set. When using the R26-ViT or self-supervised ResNet18, images were transformed to range $[-1, 1]$ to match the normalization used in the pretraining. Images were resized to 224×224 with bicubic interpolation for R26-ViT and ResNet50, and bilinear interpolation for ResNet18. During training, images were randomly flipped horizontally or vertically.

Evaluation Final models were chosen based on lowest cross entropy loss on the validation set. AUROC, F1-score (threshold=0.5), false negative rate, false positive rate, true negative rate, and true positive rate were calculated using `torchmetrics` [78]. ChampKit includes a script to perform this evaluation automatically using a single command (see Section 2.1).

3. Results

The main result of this work is the ChampKit software. ChampKit is a toolkit that enables users to evaluate many deep learning architectures on histopathology patch classification tasks. The software is reproducible, curates datasets automatically, and is easy to extend with custom datasets and models. In our experiments, ChampKit facilitated the assessment of three deep neural network architectures across six datasets, including comparing training from randomly initialized weights and transfer learning. Results for all tasks are available in Figures S1, S2, and S3. In the following subsections, we report AUROC and false negative rate as these metrics are the most suitable for these tasks. Additional metrics are reported in Supplementary Tables S2-S7.

3.1. Task 1: Identification of areas containing tumor cells

Identification of areas with tumor cells is critical in clinical histopathology. Tumor cells can have varied appearances and can be challenging to detect. In particular, small nests of tumor cells ($<$100 cells) might be difficult to detect, and this is one case where automated deep learning algorithms can be highly useful. This is especially true in sentinel lymph node biopsies, which are performed to determine whether cells from primary tumor have metastasized. False negatives are unacceptable in this situation, and so deep learning methods for this task must be rigorously evaluated. We have included detection of areas containing tumor cells as task 1 in our benchmark because of its clinical importance [79] and already wide-spread application in deep learning.

Dataset The PatchCamelyon dataset [68] is a processed and curated version of the Camelyon16 dataset [80] containing 327,680 tumor and non-tumor images at 96×96 pixels ($10\times$ magnification) from sentinel lymph node biopsies of breast cancer (Fig. 1, Fig. 2 and Table S1). An image is positively labeled if the center 32×32 pixel region contains at least one pixel of tumor. The PatchCamelyon dataset is licensed under Creative Commons Zero v1.0 Universal and is anonymized. PatchCamelyon is available for download [81] via Zenodo and is downloaded automatically when the ChampKit repository is initialized.

Results For task 1, most models performed well according to AUROC (Tables 1 and S2). Because false negatives are an unwanted outcome for tumor classification, the FNR is a more relevant metric, and R26-ViT pretrained on ImageNet had the lowest overall FNR and the highest overall AUROC. ImageNet pretraining improved performance in R26-ViT and ResNet50 but pretraining did not improve ResNet18.

3.2. Task 2: identification of areas containing tumor-infiltrating lymphocytes

Tumor-infiltrating lymphocytes (TILs) are clinically useful as prognostic biomarkers, related to the degree of immune response against a cancer. TIL quantification is important for predicting survival outcomes and guiding treatment decisions [82–85]. TILs tend to be 8–12 μm in diameter with a dark, ovoid nucleus and scant cytoplasm [86]. Despite the subtle qualitative differences of TILs across image patches, pathologists can identify TILs through visual inspection. However, in practice, they tend to characterize only a small number of microscopic fields of view [79]. More detailed prognostic patterns can be made by mapping TILs at a whole-slide-level [87]. Thus, it would be greatly beneficial to clinicians to identify areas that contain TILs across histopathology slides [36,37,84,88]. Deep learning has the potential to address major drawbacks of manual TIL scoring: inter-observer variability and the scalability of TIL detection. In response, there has been much interest in applying deep neural networks to this task [36–38,83,89,90]. Thus, task 2 consists of pan-cancer identification of regions containing TILs because of its tremendous clinical relevance and popularity in deep learning.

Dataset The dataset for task 2 consists of 304,097 TIL-positive and TIL-negative images from [65], a curated subset of the data presented in [36,37] (Fig. 3 and Table S1). This dataset includes 23 different cancer types from The Cancer Genome Atlas (TCGA) [91], representing a wide distribution of tissue types and stain differences. Patches are from formalin-fixed, paraffin-embedded (FFPE) whole slide images. Images are 100×100 pixels at 0.5 μm/pixel and are positive if they contain at least two TILs. No stain normalization was applied to the images. The data is licensed under Creative Commons Attribution 4.0 International. Images are anonymized, and there is no overlap in TCGA participants across data splits. This dataset is available for download via Zenodo

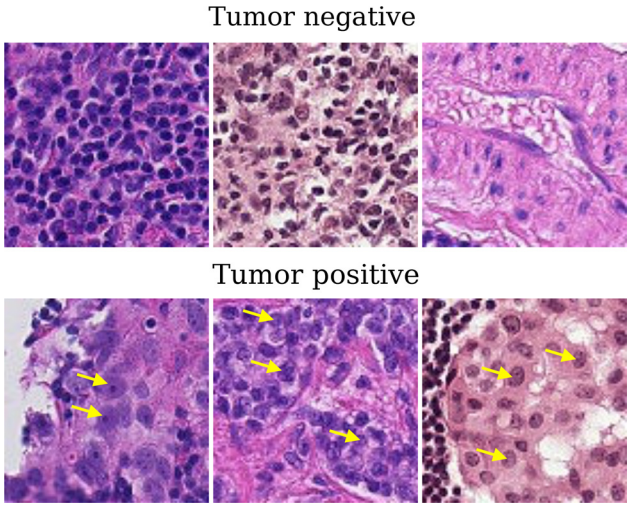


Fig. 2. Sample tumor negative and tumor positive images from PatchCamelyon dataset [68]. The yellow arrows point to examples of tumor cells. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Table 1
Results on task 1. AUROC = area under the receiver operator characteristic curve. FNR = false negative rate. Values shown are means across three runs.

Model	Pretraining	AUROC	FNR
ResNet18	None	0.929	0.243
ResNet18	ImageNet	0.933	0.250
ResNet18	SSL	0.943	0.239
ResNet50	None	0.921	0.249
ResNet50	ImageNet	0.943	0.217
R26-ViT	None	0.943	0.271
R26-ViT	ImageNet	0.962	0.191

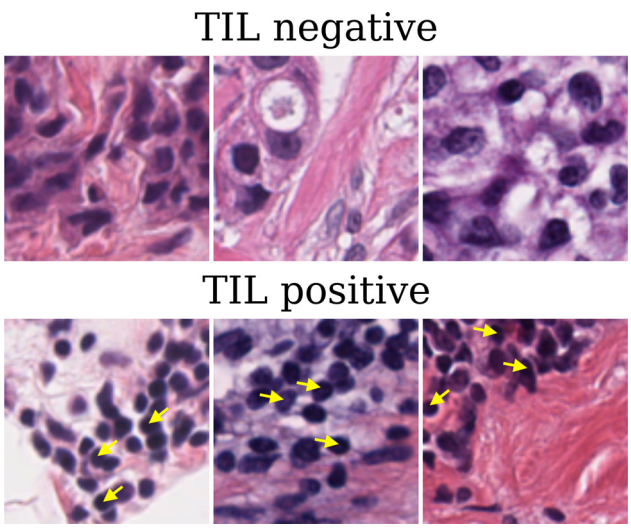


Fig. 3. Sample images from TILs dataset [65]. The yellow arrows point to examples of TILs. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Table 2
Results on task 2 (TIL detection). Values shown are means across three runs.

Model	Pretraining	AUROC	FNR
ResNet18	None	0.970	0.252
ResNet18	ImageNet	0.969	0.256
ResNet18	SSL	0.967	0.275
ResNet50	None	0.969	0.241
ResNet50	ImageNet	0.968	0.253
R26-ViT	None	0.943	0.464
R26-ViT	ImageNet	0.974	0.246

[65] and is downloaded automatically when the user initializes ChampKit.

Results In general, all of the tested models do well on task 2 (Tables 2 and S3). ResNet18 and ResNet50 are relatively consistent across pretraining strategies, though SSL pretraining resulted in slightly worse performance. The randomly initialized R26-ViT performed most poorly and had a large spread in AUROC and FNR across three runs. However, pretraining on ImageNet brought performance of R26-ViT in line with that of the ResNets, and indeed this model was best based on AUROC and FNR.

3.3. Tasks 3–5: Microsatellite instability detection

Microsatellite instability (MSI) is an important prognostic clinical biomarker and has generated strong interest in recent years. MSI causes an abundance of DNA mutations and the formation of neoantigens, which activate the immune system, and causes changes in tissue morphology [86,92,93]. MSI is a useful clinical biomarker and is an indicator for PD-1/PD-L1 blocking therapies, like pembrolizumab [94–98]. [99] recently found that their PD-1-blocking therapy led to remission in all 18 study participants. If a pathologist suspects an MSI phenotype, the standard of care is to conduct confirmatory molecular testing. Previously, [26] found that they could potentially avoid the time and cost of molecular testing by detecting MSI directly from histopathology. Many similar studies have been conducted [6,29,100–104], highlighting the importance of and excitement around MSI. We have included MSI detection in different cancer types and tissue preparations as tasks 3–5 because of the strong interest in predicting MSI from histopathology and the clinical relevance of MSI.

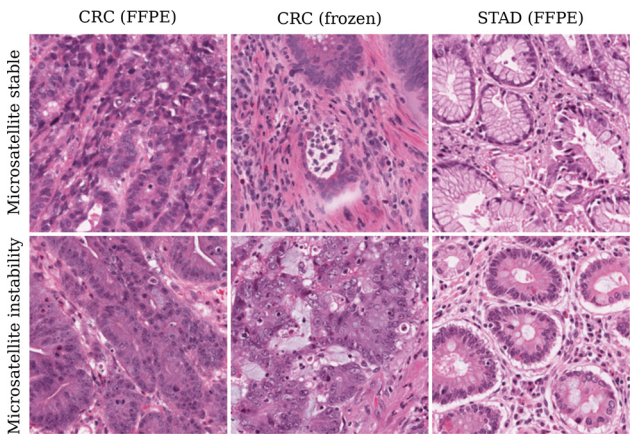


Fig. 4. Sample images from MSI datasets. Histologic features of MSI include poorly differentiated cells, signet ring cells, mucinous histopathology, cribriforming, and lymphocytic infiltrate [92]. FFPE and frozen are two different types of tissue preparations.

Dataset MSI data was curated from [26], which includes images from formalin-fixed, paraffin-embedded samples of colorectal carcinoma (CRC) and stomach adenocarcinoma (STAD) [66] and images from frozen samples of CRC [67]. All images are 224×224 pixels at 0.5 μm/pixel (Fig. 4 and Table S1). These datasets are publicly available and licensed under Creative Commons Attribution 4.0 International, and all images are anonymized. These datasets are available for immediate download via Zenodo (task 3 [66], task 4 [67], and task 5 [66]), and ChampKit automatically downloads and prepares these datasets.

Results Overall, MSI detection was the most difficult task included in ChampKit (Table 3). In Task 3, the AUROC and FNR of the ResNets were consistent across pretraining strategies (Table S4). Randomly initialized R26-ViT had the worst AUROC and but was improved significantly by ImageNet pretraining. The FNR of this model was highly variable across the three runs but was made more consistent with ImageNet pretraining. In Task 4, interestingly the randomly initialized R26-ViT had the worst AUROC but the best FNR, and pretraining with ImageNet significantly improved the AUROC but worsened the FNR (Table S5). ImageNet pretraining worsened AUROC for the ResNets, and SSL pretraining resulted in the poorest performance for ResNet18. In Task 5, ImageNet pretraining only helped ResNet50. For all other models, pretraining resulted in worse AUROC and FNR (Table S6).

3.4. Task 6: Precancerous versus benign

Colonoscopies are an important screening test for colorectal carcinoma. Polyps are commonly found during the procedure [105], and these polyps can be benign, precancerous, or cancerous. It is critical to correctly classify a benign polyp from one with cancerous potential because cancerous polyps might indicate need for additional treatment, but distinguishing between these remains challenging [106]. False negatives are unacceptable in this task, and as such, there is significant interest in using deep learning to robustly detect precancerous polyps [107–109]. Due to the clinical importance of detecting precancerous colorectal polyps and the growing interest in applying deep learning to this problem, we elected to use the MHIST dataset [64] for task 6.

Dataset This dataset includes images of hyperplastic polyps and sessile serrated adenomas (Fig. 5). Hyperplasia is a benign overgrowth of cells, and an adenoma is a precancerous, low-grade disordered growth of cells. MHIST consists of 3,152 images colorectal polyps. The images were labeled as hyperplastic or adenomas by seven pathologists, and a binary classification is made by majority

Table 3
Tasks 3–5 results. MSI detection in colorectal carconima (CRC) and stomach adenocarcinoma (STAD) with FFPE or frozen preparations. Values shown are means across three runs.

		Task 3 (CRC — FFPE)		Task 4 (CRC — frozen)		Task 5 (STAD — FFPE)	
Model	Pretraining	AUROC	FNR	AUROC	FNR	AUROC	FNR
ResNet18	None	0.661	0.628	0.708	0.632	0.710	0.555
ResNet18	ImageNet	0.666	0.665	0.698	0.657	0.693	0.582
ResNet18	SSL	0.667	0.652	0.667	0.683	0.696	0.571
ResNet50	None	0.667	0.619	0.701	0.674	0.705	0.550
ResNet50	ImageNet	0.668	0.646	0.689	0.677	0.728	0.520
R26-ViT	None	0.531	0.539	0.667	0.407	0.718	0.467
R26-ViT	ImageNet	0.676	0.697	0.732	0.656	0.709	0.586

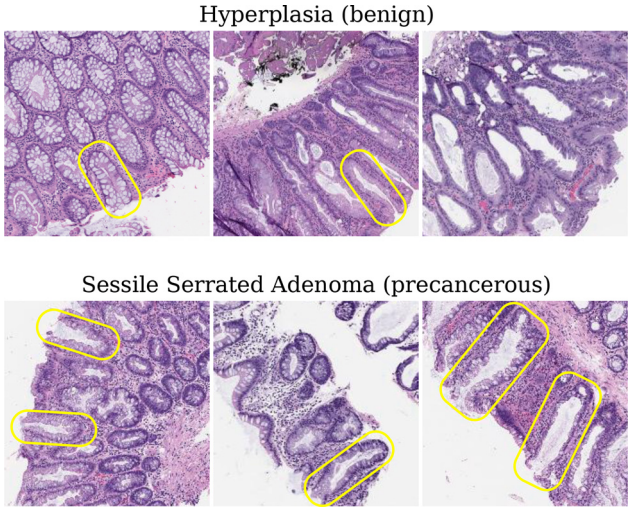


Fig. 5. Sample images from MHIST dataset [64]. Colonic crypts are outlined in yellow. Please note the difference in appearance between hyperplasia and adenoma. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Table 4
Results on task 6 (precancerous versus benign). Values shown are means across three runs.

Model	Pretraining	AUROC	FNR
ResNet18	None	0.885	0.358
ResNet18	ImageNet	0.919	0.288
ResNet18	SSL	0.906	0.298
ResNet50	None	0.823	0.596
ResNet50	ImageNet	0.921	0.220
R26-ViT	None	0.770	0.608
R26-ViT	ImageNet	0.934	0.176

vote. All images are 224×224 pixels at 8x magnification and are deidentified. The MHIST dataset is the smallest dataset included in ChampKit (Table S1), and this provides a useful test of how well different models and pretraining strategies cope with a small data regime. To access the dataset, one must complete an online form accepting a dataset use agreement. The user should then receive an automated email with a link to download the dataset. Once the dataset is downloaded, ChampKit can be used to prepare the dataset and train and evaluate models on the data.

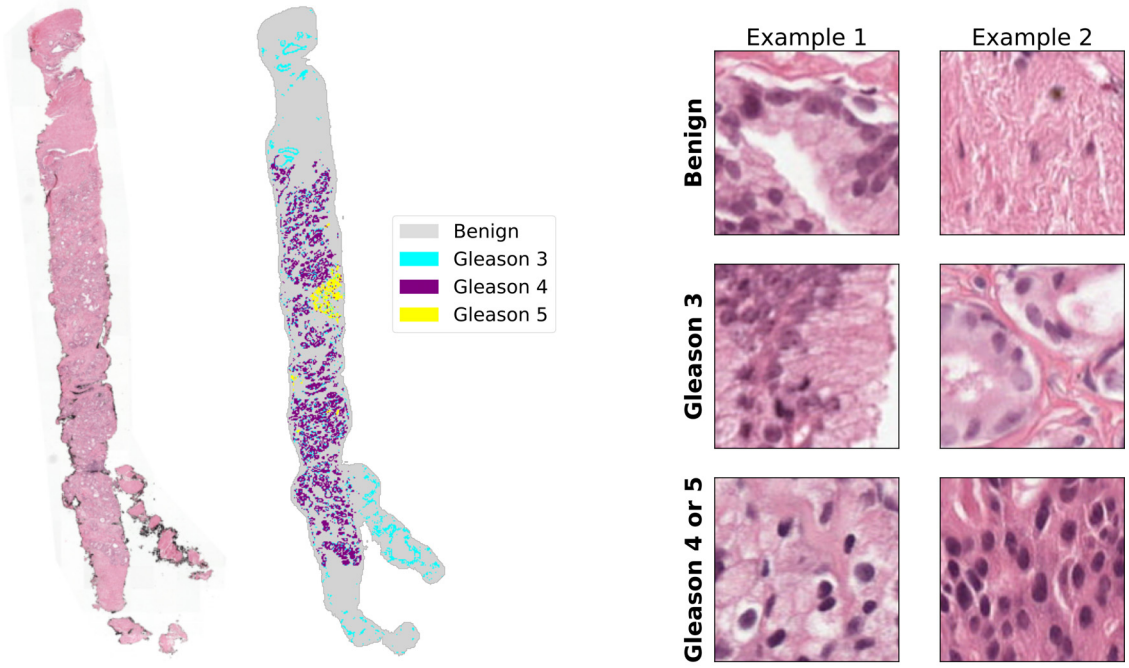
Results Unlike in the previous tasks, pretraining dramatically improved performance across all models (Tables 4 and S7), consistent with the original MHIST publication [64]. Randomly initialized R26-ViT had the worst AUROC and FNR of all models, but the ImageNet-pretrained model had the best performance overall. ResNet50 was similar in performance to R26-ViT. ResNet18 was best among the randomly initialized models, consistent with [51]. We speculate that pretraining was especially important here because of the small dataset size. Pretraining might provide useful initializations for other small datasets. SSL pretraining, however, did not provide improvements over ImageNet pretraining.

Hyperparameter search To demonstrate an evaluation of hyperparameters with ChampKit, we conducted a grid search using the ResNet18 model on Task 6, searching over the following parameters: ImageNet-pretrained (yes, no), learning rate (0.01, 0.001, 0.0001), batch size (16, 32, 64, 128), optimizer (Adam, AdamW, SGD), freeze encoder (yes, no), augmentation (yes, no; augmentation is horizontal and vertical flipping each with probabilities of 0.5), and training with automatic mixed precision (yes, no). "Freeze encoder" means that the original representation learning layers are frozen and the appended multilayer perceptron is trainable. Of the explored models, the AUROC ranged from 0.445 to 0.938, and F1 score ranged from 0.0 to 0.825 (Fig. S4). This demonstrates the effectiveness of using ChampKit to aid in hyperparameter search.

3.5. Identifying an optimal model for multi-class patch classification from annotated whole slides

To demonstrate the use of ChampKit with annotated slides and multi-class classification, we evaluated models on patches sampled from the PANDA dataset [72]. This dataset contains over 10,000 annotated biopsy images and uses the CC BY-SA-NC 4.0 license. In about half of the dataset, regions of benign epithelium and different Gleason grades are segmented (Gleason 3, 4, and 5). The biopsy images and annotations are stored as multi-resolution TIFF images (Fig. 6 a). We created a patch classification dataset using a subset of the annotated PANDA slides. A limitation of the PANDA dataset is that the Gleason segmentations are noisy, and this presents challenges when attempting to assign labels to extracted patches. We have chosen not to include the patched PANDA dataset in the ChampKit repository at this point because the noisy labels warrant further cleaning of the data. Despite the limitations, this section demonstrates 1) how one can leverage annotated whole slide images and 2) benchmark multi-class patch classification models with ChampKit.

Extracting patches from annotated whole slides In Section 2.1, we described how one can prepare a patch classification dataset from annotated whole slides, and we listed several tools. We opted to use Large Image [71] in a Python script to extract patches of 128×128 pixels at 0.5 μm/pixel. In essence, we iterated through all non-overlapping patches in an annotation image, and we kept a patch if it contained more than 10% of a label (not including background) and only one label (normal epithelium, Gleason 3, Gleason 4, or Gleason 5). The histology patch was then extracted from the associated biopsy slide (using the same coordinates as the patch in the mask image) and was saved as a PNG file in a label-specific



(a) Prostate biopsy and multi-class annotations.

(b) Examples of patches extracted from annotated regions.

Fig. 6. A sample prostate biopsy and extracted patches from the PANDA dataset [72]. This dataset includes over 10,000 biopsies and many have semantic segmentations of Gleason 3, 4, and 5, as well as normal epithelium. The annotations are noisy, as they were created using other machine learning models. For this reason, the patch labels are noisy too.

Table 5
Classification results for the patched PANDA dataset. The classes in this dataset are "benign", "Gleason3″, and "Gleason 4 or5″. Values shown are the performance for the indicated class and are means across three runs. The labels for this dataset are noisy, which could explain the high false negative rates.

		Benign		Gleason 3		Gleason 4 or 5	
Model	Pretraining	AUROC	FNR	AUROC	FNR	AUROC	FNR
ResNet18	None	0.791	0.048	0.619	1.000	0.865	0.418
ResNet18	ImageNet	0.847	0.097	0.767	0.636	0.921	0.358
ResNet18	SSL	0.853	0.095	0.776	0.660	0.919	0.331
ResNet50	None	0.796	0.044	0.626	0.998	0.870	0.414
ResNet50	ImageNet	0.835	0.076	0.769	0.695	0.906	0.366
R26-ViT	None	0.802	0.031	0.660	0.996	0.890	0.424
R26-ViT	ImageNet	0.836	0.085	0.766	0.648	0.910	0.383

directory. This was done for over 5,000 biopsies in the PANDA dataset and resulted in a total of 1,621,011 benign, 2,220 Gleason 3, 1,722 Gleason 4, and 401 Gleason 5 patches. We then removed low contrast images, because low contrast could have indicated glass or another area that we did not intend to sample. The dataset was highly unbalanced at this point. To remedy this, we merged the Gleason 4 and Gleason 5 labels into a single class, "Gleason 4 or 5", and we randomly sampled 5,000 benign patches. Please see Fig. 6 b for a sample of extracted patches and Table S8 for the final dataset size.

Results We performed the same experiment as in Tasks 1–6: ResNet18, ResNet50, and R26-ViT models were trained with or without transfer learning, and evaluation metrics were calculated on a held out test set. All experimental settings were the same except as in the previous tasks except the number of training epochs. Models were trained for a total of 50 epochs. Notably, this task uses three classes, whereas Tasks 1–6 are binary classifications. ResNet18 models pretrained with self-supervised learning or ImageNet performed best overall (Tables 5, S9–11). Models trained from random initialization performed markedly worse than transfer learning, often obtaining false negative rates close to 1.0 in Gleason 3 classification. This is consistent with results on Task 6 (MHIST), a similarly small dataset, and further suggests that transfer learning is beneficial in small data regimes. Classification performance (specifically false negative rate) varied widely among the three classes, reinforcing the need to evaluate the class-specific performance of models.

4. Discussion and conclusion

Here we introduce ChampKit, a reproducible toolkit for patch-based image classification in histopathology. We use ChampKit to provide baseline results for multiple models on six histopathology datasets. We found that transfer learning can improve classification performance, but this is not consistent across tasks. It remains unclear whether the scale of the histopathology features (i.e., magnification) plays a role in being amenable to transfer learning based on models pretrained on natural images. ChampKit enables the systematic evaluation of transfer learning on patch-based image classification, and we hope that it will greatly advance the knowledge of transfer learning and modeling innovations in histopathology. ChampKit also allows users to identify optimal classification

models for their own datasets. Our own group has used this tool to accelerate our own research. Specifically, we used it to identify the best multi-class Gleason grade classification model for a private dataset of prostate whole slide images. We trained 33 models in a hyperparameter search across several model architectures and data regularization techniques, which took approximately three days to complete. We then identified the best model using the evaluation script in ChampKit, and we applied this model to whole slides using WSInfer [110] to obtain whole-specimen maps of Gleason grade for further analysis. The combination of ChampKit and WSInfer accelerated our work and can be explored further in future work.

A major goal of ChampKit is to expedite a user's search for an ideal classification model on their digital pathology dataset. This work makes use of two main components, namely `timm` (to provide pre-trained models and training methods) and Weights and Biases (for hyperparameter search, experiment logging, and browser-based visualization of results). Other benchmarking toolkits exist, as discussed in Section 1.1, but to our knowledge, no toolkit exists specifically for patch-based classification in digital pathology. Compared to existing tools, we do not expect ChampKit to be more efficient in hyperparameter search or model training. In fact, we use existing tools to implement these features, and as they are developed, they may become more efficient. Additionally, we do not expect the models developed with ChampKit to be more accurate than models developed with other methods, in so far as a similar hyperparameter space is explored. On the other hand, the ChampKit training script is derived from `timm` and includes many options that have been used to train state-of-the-art ImageNet classifiers, and these options may also prove useful in digital pathology patch classification.

ChampKit has several limitations. There are many deep learning tasks in digital pathology, and ChampKit addresses only patch-based classification. ChampKit can be modified to support other patch-based tasks, like segmentation, though this would require the addition of segmentation-specific architectures, data loading mechanisms, loss functions, and evaluation metrics. If users would like to benchmark semantic segmentation tasks, we refer them to OpenMMLab Semantic Segmentation Toolbox and Benchmark [111]. Similarly, if users have slide-level classification tasks, we refer them to the work of Laleh et al. [50]. Additionally, it is assumed that all patches have mutually exclusive labels, though we acknowledge that it is possible that patches can potentially have multiple labels. ChampKit also does not perform pre-training on ones dataset and does not support self-supervised learning, and instead relies on previously published models for transfer learning. This limits the diversity of models that one may evaluate. We encourage the community to provide feedback, suggest features, and contribute new functionality to ChampKit via our GitHub repository.

The current study has several limitations as well. The baseline comparisons were limited to three network architectures with different pretrained weights, but this is a tiny sample of available architectures accessible to ChampKit. All models in this manuscript were also trained using identical hyperparameters for each dataset to make fair comparisons. Nevertheless, further optimization of each model could improve performance but was not explored in this study to maintain consistency. Thus, the benchmarks performed establish a lower-bound of what is achievable. As the scope of this study was to introduce the ChampKit software, a follow up study using this tool can build upon our work to elucidate modeling choices that are generalizable as part of a more comprehensive analysis.

In summary, ChampKit enables users to benchmark patch classification models and identify the best model for their dataset, which we hope will accelerate deep learning research in histopathology.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

We gratefully acknowledge support from National Cancer Institute grants U24CA215109 and UH3CA225021. JRK was also supported by National Institutes of Health grant T32GM008444 (NIGMS) and by the Medical Scientist Training Program at Stony Brook University. PKK was supported in part by funding the National Human Genome Research Institute of the National Institutes of Health under Award Number R01HG012131, the Developmental Funds from the CSHL Cancer Center Support Grant 5P30CA045508, and the Simons Center for Quantitative Biology at Cold Spring Harbor Laboratory. This work was performed with assistance from the US National Institutes of Health Grant S10OD028632-01. The results shown here are in part based upon data generated by the TCGA Research Network: https://www.cancer.gov/tcga. We thank Shushan Toneyan for her help reviewing the source code and reproducing parts of this manuscript and Satrajit S. Ghosh for help naming this project.

Supplementary material

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.cmpb.2023.107631

References

[1] R. Wightman, Pytorch image models, 2019, (https://github.com/rwightman/pytorch-image-models). 10.5281/zenodo.4414861
[2] L. Biewald, Experiment tracking with weights and biases, 2020, Software available from wandb.com, https://www.wandb.com/.
[3] S. Banerji, S. Mitra, Deep learning in histopathology: a review, Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 12 (1) (2022) e1439.
[4] J. Van der Laak, G. Litjens, F. Ciompi, Deep learning in histopathology: the path to the clinic, Nat. Med. 27 (5) (2021) 775–784.
[5] C.L. Srinidhi, O. Ciga, A.L. Martel, Deep neural network models for computational histopathology: a survey, Med Image Anal 67 (2021) 101813.
[6] A. Echle, H.I. Grabsch, P. Quirke, P.A. van den Brandt, N.P. West, G.G.A. Hutchins, L.R. Heij, X. Tan, S.D. Richman, J. Krause, et al., Clinical-grade detection of microsatellite instability in colorectal tumors by deep learning, Gastroenterology 159 (4) (2020) 1406–1416.
[7] S. Deng, X. Zhang, W. Yan, E.I. Chang, Y. Fan, M. Lai, Y. Xu, et al., Deep learning in digital pathology image analysis: a survey, Front Med 14 (4) (2020) 470–487.
[8] A.S. Sultan, M.A. Elgharib, T. Tavares, M. Jessri, J.R. Basile, The use of artificial intelligence, machine learning and deep learning in oncologic histopathology, Journal of Oral Pathology & Medicine 49 (9) (2020) 849–856.
[9] R. Gupta, T. Kurc, A. Sharma, J.S. Almeida, J. Saltz, The emergence of pathomics, Curr Pathobiol Rep 7 (3) (2019) 73–84.
[10] A. Hamidinekoo, E. Denton, A. Rampun, K. Honnor, R. Zwiggelaar, Deep learning in mammography and breast histology, an overview and future trends, Med Image Anal 47 (2018) 45–67.
[11] N. Coudray, P.S. Ocampo, T. Sakellaropoulos, N. Narula, M. Snuderl, D. Fenyö, A.L. Moreira, N. Razavian, A. Tsirigos, Classification and mutation prediction from non–small cell lung cancer histopathology images using deep learning, Nat. Med. 24 (10) (2018) 1559–1567.
[12] O. Jimenez-del Toro, S. Otálora, M. Andersson, K. Eurén, M. Hedlund, M. Rousson, H. Müller, M. Atzori, Analysis of histopathology images: from traditional machine learning to deep learning, in: Biomedical Texture Analysis, Elsevier, 2017, pp. 281–314.
[13] J. Xie, R. Liu, J. Luttrell IV, C. Zhang, Deep learning based analysis of histopathological images of breast cancer, Front Genet 10 (2019) 80.
[14] W. Mi, J. Li, Y. Guo, X. Ren, Z. Liang, T. Zhang, H. Zou, Deep learning-based multi-class classification of breast digital pathology images, Cancer Manag Res (2021) 4605–4617.
[15] M. Abdolahi, M. Salehi, I. Shokatian, R. Reiazi, Artificial intelligence in automatic classification of invasive ductal carcinoma breast cancer in digital pathology images, Med J Islam Repub Iran 34 (2020) 140.
[16] H. Yang, L. Chen, Z. Cheng, M. Yang, J. Wang, C. Lin, Y. Wang, L. Huang, Y. Chen, S. Peng, et al., Deep learning-based six-type classifier for lung cancer

and mimics from histopathological whole slide images: a retrospective study, BMC Med 19 (2021) 1–14.
[17] T. Sakamoto, T. Furukawa, K. Lami, H.H.N. Pham, W. Uegami, K. Kuroda, M. Kawai, H. Sakanashi, L.A.D. Cooper, A. Bychkov, et al., A narrative review of digital pathology and artificial intelligence: focusing on lung cancer, Transl Lung Cancer Res 9 (5) (2020) 2255.
[18] S. Wang, D.M. Yang, R. Rong, X. Zhan, J. Fujimoto, H. Liu, J. Minna, I.I. Wistuba, Y. Xie, G. Xiao, Artificial intelligence in lung cancer pathology image analysis, Cancers (Basel) 11 (11) (2019) 1673.
[19] S. Klimov, Y. Xue, A. Gertych, R.P. Graham, Y. Jiang, S. Bhattarai, S.J. Pandol, E.A. Rakha, M.D. Reid, R. Aneja, Predicting metastasis risk in pancreatic neuroendocrine tumors using deep learning image analysis, Front Oncol 10 (2021) 593211.
[20] H. Fu, W. Mi, B. Pan, Y. Guo, J. Li, R. Xu, J. Zheng, C. Zou, T. Zhang, Z. Liang, et al., Automatic pancreatic ductal adenocarcinoma detection in whole slide images using deep convolutional neural networks, Front Oncol 11 (2021) 665929.
[21] H. Le, D. Samaras, T. Kurc, R. Gupta, K. Shroyer, J. Saltz, Pancreatic cancer detection in whole slide images using noisy label annotations, in: Medical Image Computing and Computer Assisted Intervention–MICCAI 2019: 22nd International Conference, Shenzhen, China, October 13–17, 2019, Proceedings, Part I 22, Springer, 2019, pp. 541–549.
[22] M.N.M. Sehmi, M.F.A. Fauzi, W.S.H.M.W. Ahmad, E.W.L. Chan, Pancreatic cancer grading in pathological images using deep learning convolutional neural networks, F1000Res 10 (1057) (2022) 1057.
[23] H. Bowen, H. Huang, S. Zhang, D. Zhang, Q. Shi, J. Liu, J. Guo, Artificial intelligence in pancreatic cancer, Theranostics 12 (16) (2022) 6931.
[24] M. Lucas, I. Jansen, C.D. Savci-Heijink, S.L. Meijer, O.J. de Boer, T.G. van Leeuwen, D.M. de Bruin, H.A. Marquering, Deep learning for automatic Gleason pattern classification for grade group determination of prostate biopsies, Virchows Archiv 475 (2019) 77–83.
[25] Y. Tolkach, T. Dohmgörgen, M. Toma, G. Kristiansen, High-accuracy prostate cancer pathology using deep learning, Nature Machine Intelligence 2 (7) (2020) 411–418.
[26] J.N. Kather, A.T. Pearson, N. Halama, D. Jäger, J. Krause, S.H. Loosen, A. Marx, P. Boor, F. Tacke, U.P. Neumann, et al., Deep learning can predict microsatellite instability directly from histology in gastrointestinal cancer, Nat. Med. 25 (7) (2019) 1054–1056.
[27] H.S. Muti, L.R. Heij, G. Keller, M. Kohlruss, R. Langer, B. Dislich, J.-H. Cheong, Y.-W. Kim, H. Kim, M.-C. Kook, et al., Development and validation of deep learning classifiers to detect epstein-barr virus and microsatellite instability status in gastric cancer: a retrospective multicentre cohort study, The Lancet Digital Health 3 (10) (2021) e654–e664.
[28] A. Echle, N.G. Laleh, P.L. Schrammen, N.P. West, C. Trautwein, T.J. Brinker, S.B. Gruber, R.D. Buelow, P. Boor, H.I. Grabsch, et al., Deep learning for the detection of microsatellite instability from histology images in colorectal cancer: a systematic literature review, ImmunoInformatics (2021) 100008.
[29] R. Cao, F. Yang, S.-C. Ma, L. Liu, Y. Zhao, Y. Li, D.-H. Wu, T. Wang, W.-J. Lu, W.-J. Cai, et al., Development and interpretation of a pathomics-based model for the prediction of microsatellite instability in colorectal cancer, Theranostics 10 (24) (2020) 11080.
[30] Y. Liu, K. Gadepalli, M. Norouzi, G.E. Dahl, T. Kohlberger, A. Boyko, S. Venugopalan, A. Timofeev, P.Q. Nelson, G.S. Corrado, et al., Detecting cancer metastases on gigapixel pathology images, arXiv preprint arXiv:1703.02442 (2017).
[31] D. Wang, A. Khosla, R. Gargeya, H. Irshad, A.H. Beck, Deep learning for identifying metastatic breast cancer, arXiv preprint arXiv:1606.05718 (2016).
[32] B. Lee, K. Paeng, A robust and effective approach towards accurate metastasis detection and pN-stage classification in breast cancer, in: International Conference on Medical Image Computing and Computer-assisted Intervention, Springer, 2018, pp. 841–850.
[33] R. Awan, N.A. Koohbanani, M. Shaban, A. Lisowska, N. Rajpoot, Context-aware learning using transferable features for classification of breast cancer histology images, in: International Conference Image Analysis and Recognition, Springer, 2018, pp. 788–795.
[34] O. Iizuka, F. Kanavati, K. Kato, M. Rambeau, K. Arihiro, M. Tsuneki, Deep learning models for histopathological classification of gastric and colonic epithelial tumours, Sci Rep 10 (1) (2020) 1–11.
[35] S. Kwok, Multiclass classification of breast cancer in whole-slide images, in: International Conference Image Analysis and Recognition, Springer, 2018, pp. 931–940.
[36] J. Saltz, R. Gupta, L. Hou, T. Kurc, P. Singh, V. Nguyen, D. Samaras, K.R. Shroyer, T. Zhao, R. Batiste, J. Van Arnam, Spatial organization and molecular correlation of tumor-infiltrating lymphocytes using deep learning on pathology images, Cell Rep 23 (1) (2018) 181–193.
[37] S. Abousamra, R. Gupta, L. Hou, R. Batiste, T. Zhao, A. Shankar, A. Rao, C. Chen, D. Samaras, T. Kurc, J. Saltz, Deep learning-based mapping of tumor infiltrating lymphocytes in whole slide images of 23 types of cancer, Front Oncol 11 (2022), doi:10.3389/fonc.2021.806603.
[38] Z. Lu, S. Xu, W. Shao, Y. Wu, J. Zhang, Z. Han, Q. Feng, K. Huang, Deep-learning–based characterization of tumor-infiltrating lymphocytes in breast cancers from histopathology images and multiomics data, JCO Clinical Cancer Informatics 4 (2020) 480–490.
[39] N. Linder, J.C. Taylor, R. Colling, R. Pell, E. Alveyn, J. Joseph, A. Protheroe, M. Lundin, J. Lundin, C. Verrill, Deep learning for detecting tumour-infiltrating lymphocytes in testicular germ cell tumours, J. Clin. Pathol. 72 (2) (2019) 157–164.
[40] A.L.S. Meirelles, T. Kurc, J. Saltz, G. Teodoro, Effective active learning in digital pathology: acase study in tumor infiltrating lymphocytes, Comput Methods Programs Biomed 220 (2022) 106828.
[41] U. Baid, S. Pati, T.M. Kurc, R. Gupta, E. Bremer, S. Abousamra, S.P. Thakur, J.H. Saltz, S. Bakas, Federated learning for the classification of tumor infiltrating lymphocytes, arXiv preprint arXiv:2203.16622 (2022).
[42] M. Amgad, E.S. Stovgaard, E. Balslev, J. Thagaard, W. Chen, S. Dudgeon, A. Sharma, J.K. Kerner, C. Denkert, Y. Yuan, et al., Report on computational assessment of tumor infiltrating lymphocytes from the international immuno-oncology biomarker working group, npj Breast Cancer 6 (1) (2020) 1–13.
[43] J. Thiyagalingam, M. Shankar, G. Fox, T. Hey, Scientific machine learning benchmarks, Nature Reviews Physics 4 (6) (2022) 413–420.
[44] R.J. Chen, C. Chen, Y. Li, T.Y. Chen, A.D. Trister, R.G. Krishnan, F. Mahmood, Scaling vision transformers to gigapixel images via hierarchical self-supervised learning, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 16144–16155.
[45] X. Wang, S. Yang, J. Zhang, M. Wang, J. Zhang, W. Yang, J. Huang, X. Han, Transformer-based unsupervised contrastive learning for histopathological image classification, Med Image Anal 81 (2022) 102559.
[46] O. Ciga, T. Xu, A.L. Martel, Self supervised contrastive learning for digital histopathology, Machine Learning with Applications 7 (2022) 100198, doi:10.1016/j.mlwa.2021.100198.
[47] G. Fursin, Invited talk abstract: Introducing requEST: An open platform for reproducible and quality-efficient systems-ML tournaments, 2018 1st Workshop on Energy Efficient Machine Learning and Cognitive Computing for Embedded Applications (EMC2), IEEE, 2018. 3–3
[48] J. Thiyagalingam, K. Leng, S. Jackson, J. Papay, M. Shankar, G. Fox, T. Hey, SciMLBench: A benchmarking suite for AI for science, 2021, https://github.com/stfc-sciml/sciml-bench.
[49] A. Narayan, P. Molino, K. Goel, W. Neiswanger, C. Ré, Personalized benchmarking with the ludwig benchmarking toolkit, arXiv preprint arXiv:2111.04260 (2021).
[50] N.G. Laleh, H.S. Muti, C.M.L. Loeffler, A. Echle, O.L. Saldanha, F. Mahmood, M.Y. Lu, C. Trautwein, R. Langer, B. Dislich, R.D. Buelow, H.I. Grabsch, H. Brenner, J. Chang-Claude, E. Alwers, T.J. Brinker, F. Khader, D. Truhn, N.T. Gaisa, P. Boor, M. Hoffmeister, V. Schulz, J.N. Kather, Benchmarking weakly-supervised deep learning pipelines for whole slide classification in computational pathology, Med Image Anal 79 (2022) 102474.
[51] Y. Sharma, L. Ehsany, S. Syed, D.E. Brown, Histotransfer: Understanding transfer learning for histopathology, in: 2021 IEEE EMBS International Conference on Biomedical and Health Informatics (BHI), IEEE, 2021, pp. 1–4.
[52] S. Kornblith, J. Shlens, Q.V. Le, Do better imagenet models transfer better? in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2019, pp. 2661–2671.
[53] M. Raghu, C. Zhang, J. Kleinberg, S. Bengio, Transfusion: understanding transfer learning for medical imaging, Adv Neural Inf Process Syst 32 (2019).
[54] M. Barker, N.P. Chue Hong, D.S. Katz, A.-L. Lamprecht, C. Martinez-Ortiz, F. Psomopoulos, J. Harrow, L.J. Castro, M. Gruenpeter, P.A. Martinez, et al., Introducing the FAIR principles for research software, Sci Data 9 (1) (2022) 1–6.
[55] M.D. Wilkinson, M. Dumontier, I.J. Aalbersberg, G. Appleton, M. Axton, A. Baak, N. Blomberg, J.-W. Boiten, L.B. da Silva Santos, P.E. Bourne, et al., The FAIR guiding principles for scientific data management and stewardship, Sci Data 3 (1) (2016) 1–9.
[56] U. Rubens, R. Mormont, L. Paavolainen, V. Bäcker, B. Pavie, L.A. Scholz, G. Michiels, M. Maška, D. Ünay, G. Ball, R. Hoyoux, R. Vandaele, O. Golani, S.G. Stanciu, N. Sladoje, P. Paul-Gilloteaux, R. Marée, S. Tosi, Biaflows: a collaborative framework to reproducibly deploy and benchmark bioimage analysis workflows, Patterns 1 (3) (2020) 100040, doi:10.1016/j.patter.2020.100040.
[57] B. Bischl, G. Casalicchio, M. Feurer, P. Gijsbers, F. Hutter, M. Lang, R.G. Mantovani, J.N. van Rijn, J. Vanschoren, Openml benchmarking suites, arXiv preprint arXiv:1708.03731 (2017).
[58] S.R. Piccolo, T.J. Lee, E. Suh, K. Hill, ShinyLearner: a containerized benchmarking tool for machine-learning classification of tabular data, Gigascience 9 (4) (2020), doi:10.1093/gigascience/giaa026. Giaa026
[59] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A.C. Berg, L. Fei-Fei, ImageNet large scale visual recognition challenge, International Journal of Computer Vision (IJCV) 115 (3) (2015) 211–252, doi:10.1007/s11263-015-0816-y.
[60] O. Ciga, Native pytorch weights (trained with 400 thousand images), 2022, (https://github.com/ozanciga/self-supervised-histopathology/releases/tag/nativetenpercent).
[61] T. maintainers, contributors, TorchVision: PyTorch's Computer Vision library, 2016, https://github.com/pytorch/vision.
[62] J. Bergstra, Y. Bengio, Random search for hyper-parameter optimization, Journal of machine learning research 13 (2) (2012).
[63] R. Turner, D. Eriksson, M. McCourt, J. Kiili, E. Laaksonen, Z. Xu, I. Guyon, Bayesian optimization is superior to random search for machine learning hyperparameter tuning: Analysis of the black-box optimization challenge 2020, in: NeurIPS 2020 Competition and Demonstration Track, PMLR, 2021, pp. 3–26.
[64] J. Wei, A. Suriawinata, B. Ren, X. Liu, M. Lisovsky, L. Vaickus, C. Brown, M. Baker, N. Tomita, L. Torresani, J. Wei, S. Hassanpour, A petri dish for histopathology image analysis, in: International Conference on Artificial Intelligence in Medicine, Springer, 2021, pp. 11–24. https://doi.org/10.1007/978-3-030-77211-6_2

[65] J.R. Kaczmarzyk, S. Abousamra, T. Kurc, R. Gupta, J. Saltz, Dataset for tumor infiltrating lymphocyte classification (304,097 images from TCGA), 2022, 10.5281/zenodo.6604094

[66] J.N. Kather, Histological images for MSI vs. MSS classification in gastrointestinal cancer, FFPE samples, 2019a, 10.5281/zenodo.2530835

[67] J.N. Kather, Histological images for MSI vs. MSS classification in gastrointestinal cancer, snap-frozen samples, 2019b, 10.5281/zenodo.2532612

[68] B.S. Veeling, J. Linmans, J. Winkens, T. Cohen, M. Welling, Rotation equivariant CNNs for digital pathology, in: International Conference on Medical image computing and computer-assisted intervention, Springer, 2018, pp. 210–218. https://doi.org/10.1007/978-3-030-00934-2_24

[69] P. Bankhead, M.B. Loughrey, J.A. Fernández, Y. Dombrowski, D.G. McArt, P.D. Dunne, S. McQuaid, R.T. Gray, L.J. Murray, H.G. Coleman, et al., Qupath: open source software for digital pathology image analysis, Sci Rep 7 (1) (2017) 1–7.

[70] A. Goode, B. Gilbert, J. Harkes, D. Jukic, M. Satyanarayanan, Openslide: a vendor-neutral software foundation for digital pathology, J Pathol Inform 4 (1) (2013) 27.

[71] L.I. maintainers, contributors, Large Image: Python modules to work with large multiresolution images, 2019, https://github.com/girder/large_image.

[72] W. Bulten, K. Kartasalo, P.-H.C. Chen, P. Ström, H. Pinckaers, K. Nagpal, Y. Cai, D.F. Steiner, H. van Boven, R. Vink, et al., Artificial intelligence for diagnosis and gleason grading of prostate cancer: the PANDA challenge, Nat. Med. 28 (1) (2022) 154–163.

[73] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, S. Chintala, PyTorch: an imperative style, high-performance deep learning library, in: H. Wallach, H. Larochelle, A. Beygelzimer, F.d. Alché-Buc, E. Fox, R. Garnett (Eds.), Advances in Neural Information Processing Systems 32, Curran Associates, Inc., 2019, pp. 8024–8035. http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf

[74] B.J. Heil, M.M. Hoffman, F. Markowetz, S.-I. Lee, C.S. Greene, S.C. Hicks, Reproducibility standards for machine learning in the life sciences, Nat. Methods 18 (10) (2021) 1132–1135.

[75] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.

[76] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al., An image is worth 16x16 words: transformers for image recognition at scale, arXiv preprint arXiv:2010.11929 (2020).

[77] I. Loshchilov, F. Hutter, Decoupled weight decay regularization, arXiv preprint arXiv:1711.05101 (2017).

[78] N.S. Detlefsen, J. Borovec, J. Schock, A. Harsh, T. Koker, L.D. Liello, D. Stancl, C. Quan, M. Grechkin, W. Falcon, TorchMetrics - Measuring Reproducibility in PyTorch, 2022, 10.21105/joss.04101

[79] P.L. Fitzgibbons, J.L. Connolly, College of American Pathologists, Protocol for the examination of resection specimens from patients with invasive carcinoma of the breast (version 4.6.0.0), 2022, https://documents.cap.org/protocols/Breast.Invasive_4.6.0.0.REL_CAPCP.pdf.

[80] B.E. Bejnordi, M. Veta, P.J. Van Diest, B. Van Ginneken, N. Karssemeijer, G. Litjens, J.A. Van Der Laak, M. Hermsen, Q.F. Manson, M. Balkenhol, et al., Diagnostic assessment of deep learning algorithms for detection of lymph node metastases in women with breast cancer, JAMA 318 (22) (2017) 2199–2210.

[81] B.S. Veeling, J. Linmans, J. Winkens, T. Cohen, M. Welling, Rotation Equivariant CNNs for Digital Pathology, 2018, https://zenodo.org/record/2546921.

[82] G.E. Idos, J. Kwok, N. Bonthala, L. Kysh, S.B. Gruber, C. Qu, The prognostic implications of tumor infiltrating lymphocytes in colorectal cancer: a systematic review and meta-analysis, Sci Rep 10 (1) (2020) 1–14.

[83] S.T. Paijens, A. Vledder, M. de Bruyn, H.W. Nijman, Tumor-infiltrating lymphocytes in the immunotherapy era, Cellular & Molecular Immunology 18 (4) (2021) 842–859.

[84] F. Pagès, B. Mlecnik, F. Marliot, G. Bindea, F.-S. Ou, C. Bifulco, A. Lugli, I. Zlobec, T.T. Rau, M.D. Berger, et al., International validation of the consensus immunoscore for the classification of colon cancer: a prognostic and accuracy study, The Lancet 391 (10135) (2018) 2128–2139.

[85] M. Shaban, S.A. Khurram, M.M. Fraz, N. Alsubaie, I. Masood, S. Mushtaq, M. Hassan, A. Loya, N.M. Rajpoot, A novel digital score for abundance of tumour infiltrating lymphocytes predicts disease free survival in oral squamous cell carcinoma, Sci Rep 9 (1) (2019) 1–13.

[86] V. Kumar, A.K. Abbas, J.C. Aster, Robbins Basic Pathology, Elsevier, Philadelphia, 2017.

[87] D.J. Fassler, L.A. Torre-Healy, R. Gupta, A.M. Hamilton, S. Kobayashi, S.C. Van Alsten, Y. Zhang, T. Kurc, R.A. Moffitt, M.A. Troester, K.A. Hoadley, J. Saltz, Spatial characterization of tumor-infiltrating lymphocytes and breast cancer progression, Cancers (Basel) 14 (9) (2022) 2148, doi:10.3390/cancers14092148.

[88] R. Salgado, C. Denkert, S. Demaria, N. Sirtaine, F. Klauschen, G. Pruneri, S. Wienert, G. Van den Eynden, F.L. Baehner, F. Pénault-Llorca, et al., The evaluation of tumor-infiltrating lymphocytes (TILs) in breast cancer: recommendations by an international TILs working group 2014, Annals of Oncology 26 (2) (2015) 259–271.

[89] H. Le, R. Gupta, L. Hou, S. Abousamra, D. Fassler, L. Torre-Healy, R.A. Moffitt, T. Kurc, D. Samaras, R. Batiste, T. Zhao, A. Rao, A.L. Van Dyke, A. Sharma, E. Bremer, J.S. Almeida, J. Saltz, Utilizing automated breast cancer detection to identify spatial distributions of tumor-infiltrating lymphocytes in invasive breast cancer, Am. J. Pathol. 190 (7) (2020) 1491–1504.

[90] X. Zhang, X. Zhu, K. Tang, Y. Zhao, Z. Lu, Q. Feng, DdtNet: a dense dual-task network for tumor-infiltrating lymphocyte detection and segmentation in histopathological images of breast cancer, Med Image Anal 78 (2022) 102415, doi:10.1016/j.media.2022.102415.

[91] K.A. Hoadley, C. Yau, T. Hinoue, D.M. Wolf, A.J. Lazar, E. Drill, R. Shen, A.M. Taylor, A.D. Cherniack, V. Thorsson, et al., Cell-of-origin patterns dominate the molecular classification of 10,000 tumors from 33 types of cancer, Cell 173 (2) (2018) 291–304.

[92] J. Alexander, T. Watanabe, T.-T. Wu, A. Rashid, S. Li, S.R. Hamilton, Histopathological identification of colon cancer with microsatellite instability, Am. J. Pathol. 158 (2) (2001) 527–535.

[93] G. Germano, S. Lamba, G. Rospo, L. Barault, A. Magrì, F. Maione, M. Russo, G. Crisafulli, A. Bartolini, G. Lerda, et al., Inactivation of DNA repair triggers neoantigen generation and impairs tumour growth, Nature 552 (7683) (2017) 116–120.

[94] S.J. Casak, L. Marcus, L. Fashoyin-Aje, S.L. Mushti, J. Cheng, Y.-L. Shen, W.F. Pierce, L. Her, K.B. Goldberg, M.R. Theoret, et al., FDA approval summary: pembrolizumab for the first-line treatment of patients with MSI-H/dMMR advanced unresectable or metastatic colorectal carcinoma, Clinical Cancer Research 27 (17) (2021) 4680–4684.

[95] D.M. O'Malley, G.M. Bariani, P.A. Cassier, A. Marabelle, A.R. Hansen, A. De Jesus Acosta, W.H. Miller, T. Safra, A. Italiano, L. Mileshkin, et al., Pembrolizumab in patients with microsatellite instability–high advanced endometrial cancer: results from the KEYNOTE-158 study, Journal of Clinical Oncology 40 (7) (2022) 752–761.

[96] C. Luchini, F. Bibeau, M. Ligtenberg, N. Singh, A. Nottegar, T. Bosse, R. Miller, N. Riaz, J.-Y. Douillard, F. Andre, et al., Esmo recommendations on microsatellite instability testing for immunotherapy in cancer, and its relationship with pd-1/pd-l1 expression and tumour mutational burden: a systematic review-based approach, Annals of Oncology 30 (8) (2019) 1232–1243.

[97] F. Pietrantonio, G. Randon, M. Di Bartolomeo, A. Luciani, J. Chao, E.C. Smyth, F. Petrelli, Predictive role of microsatellite instability for PD-1 blockade in patients with advanced gastric cancer: a meta-analysis of randomized clinical trials, ESMO open 6 (1) (2021) 100036.

[98] L.A. Diaz, D.T. Le, T. Yoshino, T. Andre, J.C. Bendell, M. Rosales, S.P. Kang, B. Lam, D. Jäger, Keynote-177: Phase 3, open-label, randomized study of first-line pembrolizumab (pembro) versus investigator-choice chemotherapy for mismatch repair-deficient (dmmr) or microsatellite instability-high (msi-h) metastatic colorectal carc@inproceedingspmlr-v139-touvron21a, title = Training data-efficient image transformers & distillation through attention, author = Touvron, Hugo and Cord, Matthieu and Douze, Matthijs and Massa, Francisco and Sablayrolles, Alexandre and Jegou, Herve, booktitle = International Conference on Machine Learning, pages = 10347–10357, year = 2021, volume = 139, month = Julyinoma (mcrc), 2018,

[99] A. Cercek, M. Lumish, J. Sinopoli, J. Weiss, J. Shia, M. Lamendola-Essel, I.H. El Dika, N. Segal, M. Shcherba, R. Sugarman, Z. Stadler, R. Yaeger, J.J. Smith, B. Rousseau, G. Argiles, M. Patel, A. Desai, L.B. Saltz, M. Widmar, K. Iyer, J. Zhang, N. Gianino, C. Crane, P.B. Romesser, E.P. Pappou, P. Paty, J. Garcia-Aguilar, M. Gonen, M. Gollub, M.R. Weiser, K.A. Schalper, L.A. Diaz, Pd-1 blockade in mismatch repairdeficient, locally advanced rectal cancer, N top N. Engl. J. Med. (2022), doi:10.1056/NEJMoa2201445.

[100] R. Yamashita, J. Long, T. Longacre, L. Peng, G. Berry, B. Martin, J. Higgins, D.L. Rubin, J. Shen, Deep learning model for the prediction of microsatellite instability in colorectal cancer: a diagnostic study, The Lancet Oncology 22 (1) (2021) 132–141.

[101] M.A. Jenkins, S. Hayashi, A.-M. O'shea, L.J. Burgart, T.C. Smyrk, D. Shimizu, P.M. Waring, A.R. Ruszkiewicz, A.F. Pollett, M. Redston, et al., Pathology features in bethesda guidelines predict colorectal cancer microsatellite instability: a population-based study, Gastroenterology 133 (1) (2007) 48–56.

[102] J. Shia, N.A. Ellis, P.B. Paty, G.M. Nash, J. Qin, K. Offit, X.-M. Zhang, A.J. Markowitz, K. Nafa, J.G. Guillem, et al., Value of histopathology in predicting microsatellite instability in hereditary nonpolyposis colorectal cancer and sporadic colorectal cancer, Am. J. Surg. Pathol. 27 (11) (2003) 1407–1417.

[103] A. Hyde, D. Fontaine, S. Stuckless, R. Green, A. Pollett, M. Simms, P. Sipahimalani, P. Parfrey, B. Younghusband, A histology-based model for predicting microsatellite instability in colorectal cancers, Am. J. Surg. Pathol. 34 (12) (2010) 1820–1829.

[104] M.R. Alam, J. Abdul-Ghafar, K. Yim, N. Thakur, S.H. Lee, H.-J. Jang, C.K. Jung, Y. Chong, Recent applications of artificial intelligence from histopathologic image-based prediction of microsatellite instability in solid cancers: a systematic review, Cancers (Basel) 14 (11) (2022) 2590.

[105] J.C. Obuch, C.M. Pigott, D.J. Ahnen, Sessile serrated polyps: detection, eradication, and prevention of the evil twin, Curr Treat Options Gastroenterol 13 (1) (2015) 156–170.

[106] D.R. Jaravaza, J.M. Rigby, Hyperplastic polyp or sessile serrated lesion? the contribution of serial sections to reclassification, Diagn Pathol 15 (1) (2020) 1–9.

[107] D. Yoon, H.-J. Kong, B.S. Kim, W.S. Cho, J.C. Lee, M. Cho, M.H. Lim, S.Y. Yang, S.H. Lim, J. Lee, et al., Colonoscopic image synthesis with generative adversarial network for enhanced detection of sessile serrated lesions using convolutional neural network, Sci Rep 12 (1) (2022) 1–12.

[108] J.W. Wei, A.A. Suriawinata, L.J. Vaickus, B. Ren, X. Liu, M. Lisovsky, N. Tomita, B. Abdollahi, A.S. Kim, D.C. Snover, et al., Evaluation of a deep neural network for automated classification of colorectal polyps on histopathologic slides, JAMA network open 3 (4) (2020). e203398–e203398

[109] B. Korbar, A.M. Olofson, A.P. Miraflor, C.M. Nicka, M.A. Suriawinata, L. Torresani, A.A. Suriawinata, S. Hassanpour, Deep learning for classification of colorectal polyps on whole-slide images, J Pathol Inform 8 (2017).

[110] J.R. Kaczmarzyk, T.M. Kurc, J.H. Saltz, WSInfer: blazingly fast pipeline for patch-based classification in whole slide images, 2022. https://github.com/SBU-BMI/wsinfer.

[111] M. Contributors, MMSegmentation: Openmmlab semantic segmentation toolbox and benchmark, 2020, (https://github.com/open-mmlab/mmsegmentation).