Article

A whole slide image managing library based on fastai for deep learning in the context of histopathology: Two use-cases explained

Christoph Neuner[1], Roland Coras[1], Ingmar Blümcke[1], Alexander Popp[1], Sven M. Schlaffer[2], Michael Buchfelder[2] and Samir Jabari[1]

[1] Institute of Neuropathology, University Hospital, Erlangen, Germany
[2] Department of Neurosurgery, University Hospital, Erlangen, Germany
* **Correspondence:** Samir Jabari, Prof. Dr. med., University Hospital Erlangen, Institute of Neuropathology Schwabachanlage 6, 91054 Erlangen Germany; Email: samir.jabari@fau.de

Abstract: Background: Processing whole-slide images (WSI) to train neural networks can be intricate and laborious. We developed an open-source library covering recurrent tasks in processing of WSI and in evaluating the performance of the trained networks for classification tasks. **Methods:** Two histopathology use-cases were selected. First we aimed to train a CNN to distinguish H&E-stained slides obtained from neuropathologically classified low-grade epilepsy-associated dysembryoplastic neuroepithelial tumor (DNET) and ganglioglioma (GG). The second project we trained a convolutional neural network (CNN) to predict the hormone expression of pituitary adenoms only from hematoxylin and eosin (H&E) stained slides. In the same approach, we addressed the issue to also predict clinically silent corticotroph adenoma. We included four clinico-pathological disease conditions in a multilabel approach. **Results:** Our best performing CNN achieved an area under the curve (AUC) of 0.97 for the receiver operating characteristic (ROC) for corticotroph adenoma, 0.86 for silent corticotroph adenoma and 0.98 for gonadotroph adenoma. Our DNET-GG classifier achieved an AUC of 1.00 for the ROC curve. All scores were calculated with the help of our library on predictions on a case basis. **Conclusions:** Our comprehensive library is most helpful to standardize the work-flow and minimize the work-burden in training CNN. It is also compatible with fastai. Indeed, our new CNNs reliably extracted neuropathologically relevant information from the H&E staining only. This approach will supplement the clinico-pathological diagnosis of brain tumors, which is currently based on cost-intense microscopic examination and variable panels of immunohistochemical stainings.

Keywords: brain, pituitary adenoma, Dysembryoplastic neuroepithelial tumor, DNET, Ganglioglioma, deep learning, digital pathology, convolutional neural network, computer vision, machine learning, convolutional neural network, CNN

1. Introduction

With increasing availability of digital microscopy scanners and whole slide imaging, digital pathology (DP) will continue to successfully implement into our daily routine diagnostic practice. These digitized slides provide the intriguing opportunity to also apply image analysis techniques for advanced applications, such as disease classification. Deep learning (DL) is the most commonly applied technology in the realm of feature learning, and iteratively improves learned representations of the region-of-interest in order to achieve maximum class separability. Medical and non-medical image-classification tasks have been remarkably successful utilizing DL. Successful examples range from utilization of different types of cancer detection/classification/grading [1,2], Focal Cortical Dysplasia in human focal epilepsies [3], classification of liver cirrhosis, heart failure detection, and classification of Alzheimer plaques [4]. Disease grading, prognosis prediction and imaging biomarkers for genetic subtype identification are more challenging tasks but have also been successfully established [5,6].

The area of computational image analysis of DP images has been addressed already by many previous works. A prerequisite to successfully apply deep learning requires domain-associated knowledge in the field of DL and DP. Whereas many pathologists are not familiar with the problem-specific tasks and technical issues for applying DL techniques, DL developers most often have little experience with histology and histopathology-associated workflows. In addition, currently available open-source tools and tutorials do not provide guidance for the needs of both groups and available programming libraries and tools (either open- or closed-source) are not targeted for an application by a pathologist or clinician with little experience into DL programming routine. This is a major obstacle for researchers to use or extend the available technology and investigate their clinical use-case and hypotheses. We developed, therefore, an open-source library specifically tuned and adjusted to the special needs of digital pathology-associated analysis tasks in the context of DL. We showcase the potential of our library by outlining two specific projects, each driven by a unique clinical hypotheses.

1.1. Use case 1: Classifying low-grade epilepsy-associated brain tumors

Dysembryoplastic neuroepithelial tumor (DNET) and ganglioglioma (GG) are slowly growing tumors composed of both, glial and neuronal cell elements, and histopathologically often difficult to classify [7]. They account for 1-2% of all brain tumors and do not metastasize or spread beyond the primary site of origin. These tumors occur mainly in children and young adults with long-standing drug-resistant epilepsy. The average age at seizure onset was 12 years in 984 GG and 14 years in 565 DNET when reviewing a large European cohort of 9523 patients who underwent epilepsy surgery. Seizures are commonly focal with or without secondary generalization, and neurosurgical resection has proven as most successful treatment option. Malignant transformation has been reported for the group of GG [8,9] whereas DNET almost never show this behavior [10]. Therefore, a precise histopathological diagnosis and differentiation of these two tumor entities is important for clinical patient management [11]. The problem is that even in specialized medical centers the inter-rater agreement on the diagnosis accounts for only 40% of these tumors [7]. The DL task was to develop, therefore, a binary classifier distinguishing between the two entities.

1.2. Use case 2: Prediction of pituitary adenoma subtypes and their neuroendocrine features

Better neuroimaging techniques and diagnostic modalities recognize more pituitary adenomas than previously thought [12]. We consider three clinical subclasses: Pituitary adenomas with A. prominent neuroendocrine symptoms, B. slowly developing, insidious, non-specific complaints delaying accurate diagnosis, or C. incidentally detected adenomas being symptom-free. It remains, therefore, challenging to accurately determine the prevalence and incidence of pituitary adenomas in the general population. They account for 15 % of all intracranial neoplasms, being the third most frequent tumor type after meningiomas and gliomas. In multiple postmortem studies, the mean prevalence of pituitary adenomas was 14.4% [12]. The overall estimated prevalence of pituitary adenomas in the general population was calculated as 16.7 %. Radiography studies showed a higher prevalence of 22.5% (Aflorei & Korbonits, 2014)(Ezzat et al., 2004). The tumor has its maximum age frequency in patients between 40 and 60 years of age. The frequency of different subtypes varies depending on the age and gender of the patients [13].

The WHO classification of pituitary adenoma from 2017 is based mainly on the hormone and transcription factor expression of the adenoma cells [14]. In common routine workup for adenomas of the pituitary gland, the morphological evaluation is based, therefore, on H&E and a panel of immunohistochemical staining for all pituitary hormones (adrenocorticotropic hormone (ACTH), luteinizing hormone (LH), follicle-stimulating hormone (FSH), prolactin (PRL), thyroid stimulating hormone (TSH), somatotropic hormone (STH)) and transcription factors. In our study, we focused on corticotroph and gonadotroph adenomas since they represent the most common subtypes. We labeled our tumor samples of corticotroph and gonadotroph adenomas accordingly, e.g. corticotroph adenoma, gonadotroph adenoma with the expression of LH, and gonadotroph adenoma with the expression of FSH. As adenomas are often non-exclusively positive for only one hormone, many cases received more than one label. Therefore we chose to tackle the problem as a multilabel approach, which means that the different classes are rated and scored individually, and possible correlations must be learned by the CNN. To make sure that the labels are correct for each tile, we manually reviewed the extracted regions from the H&E slides with the corresponding regions in the immunohistochemically stained images. In addition, we included those corticotroph adenomas as a separate class, in which the patient does not show clinical symptoms of Morbus Cushing (silent corticotropic adenoma). The DL tasks were to classify entities of adenomas of the pituitary gland from H&E-stained slides as well as to predict the clinical parameter of asymptomatic or clinically silent corticotroph adenomas.

2. Materials and Methods

2.1. The library

Compared to common image datasets consisting of small files in e.g. PNG or TIFF format WSI provide more challenges in the context of training a neural network with them. First there is the size. A WSI's typical size in the realm of Neuropathology is 0.5 – 3 Gbyte. Therefore, it is impossible to feed an entire WSI let alone a batch of WSI into a CNN, since graphic processing units or graphic cards (GPUs) do not have enough memory. So WSI need to be divided into smaller images usually referred to as tiles. WSI are also stored in special file types and most WSI scanner manufacturer provide their own. Usually, WSI are also not independent of each other. A WSI belongs to a case and a case belongs to a patient. This is important for the dataset split and evaluation of the model after the training. It is common practice to not mix data from one patient in the training, validation and test set. For evaluation it is interesting how the model performs on tile level, but usually the performance on WSI, case or patient level has a higher value in practice. So these connections need to be tracked throughout the whole process from preprocessing until postprocessing/evaluation. Our library [15] is meant to help with this common overhead in preprocessing and the evaluation for training a classification model with WSI.

2.2.Tile calculation

So the first step is to split up a WSI into multiple small tiles. A complete sample pipeline can be found in the github repository of the library (https://github.com/FAU-DLM/wsi_processing_pipeline/tree/master/tile_extraction/example.ipynb) and in the repositories of the the two use cases (https://github.com/ChristophNeuner/DNET_vs_Ganglioglioma/blob/main/dnet_vs_gg.ipynb and (https://github.com/ChristophNeuner/glioblastoma_methylation/blob/master/methylation_status_binary_classification.ipynb).

Usually not all parts of a WSI are of interest for further processing. So in general there are two main ways of making sure only the relevant parts are used: Marking the interesting regions manually or using some sort of filtering algorithms that e.g. distinguish tissue from background, filter out pencil markings or blurred tissue. Both ways are supported by the library and will be further explained in the following lines.

2.3 Filters applied on complete WSI

Our library originated as a fork of Deron Eriksson's github repository "python-wsi-preprocessing" (https://github.com/deroneriksson/python-wsi-preprocessing), which was originally written and used for his and his team's pariticipation in the Tumor Proliferation Assessment Challenge 2016 (TUPAC16) [16].

Most parts of this library have gotten a substantial rewrite and many additions were made since. However, the filters were mostly kept untouched. A great documentation about them can be found in Deron Erikson's github repository [17]: https://github.com/deroneriksson/python-wsi-preprocessing/blob/master/docs/wsi-preprocessing-in-python/index.md#apply-filters-for-tissue-segmentation.

2.4 Calculation of tile locations

Our preferred way of defining polygonal regions of interest (ROIs) in a WSI is to use the program QuPath [18] (Supplement 7). The next step is to extract the coordinates of the polygons' vertices. We wrote a small QuPath script which can be used in the "Automate" Tab in QuPath and exports the polygons' vertices' coordinates into a json file. The script can be found here:https://github.com/FAU-DLM/wsi_processing_pipeline/blob/master/QuPath_scripts/polygon_points_to_json.groovy

The next step is to convert this information into RegionOfInterestPolygon objects (https://github.com/FAU-DLM/wsi_processing_pipeline/blob/master/shared/roi.py#L66). There is a convenience function if the ROIs were annotated and extracted with our script from QuPath. This function can be found here: https://github.com/FAU-DLM/wsi_processing_pipeline/blob/master/shared/roi.py#L195

It is important to notice that this part is completely optional. You do not have to define any ROIs.

Subsequently all relevant tile locations are calculated. For this process the function "WsisToTilesParallel" (https://github.com/FAU-DLM/wsi_processing_pipeline/blob/8c5e4a360fa369221ce86dd35837e91f31817d30/tile_extraction/tiles.py#L1275) is used. It basically calls the function "WsiToTiles" for every WSI and runs in parallel. It takes a few interesting parameters. We will elaborate on the few most interesting here, the rest is covered in the function's doc string.

"wsi_paths":

First of all a list with the paths to the WSI files has to be passed. Notice that not only WSI files but also PNG files are supported here. So If one has already extracted the interesting parts of the WSI as PNGs, one can use them and do not have to specify ROI coordinates like described before.

"grids_per_roi", "optimize_grid_angles", "angle_stepsize", "minimal_tile_roi_intersection_ratio":

The library lays a grid of all possible tiles over each ROI (Supplement 8). If no ROI is specified, the library internally creates one ROI, which simply spans the complete WSI.

The logic for this part of the pipeline resides in the tiles.py module. To be more specific in the Vertex, Rectangle, Grid and GridManager classes. A Vertex object represents one vertex of the polygonal ROI and provides simple arithmetic operations like add, subtract and multiply with scalars and matrices. It also provides the functionality to rotate itself around a specified point. This is done by multiplying a rotation matrix with the vertex coordinates represented as a 2x1 vector.

$$Rotation\ Matrix \begin{matrix} x' \\ y' \end{matrix} = \begin{bmatrix} cos(\alpha) & -sin(\alpha) \\ sin(\alpha) & cos(\alpha) \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

The Vertex class also provides a convenience function to change the WSI level of the coordinates. Because of its size a WSI is stored in a pyramid like format (Supplement 10) in multiple images per level. So only if you zoom in the images of that particular region are loaded with higher resolution. Therefore during the process of tile calculations it is important to specifiy the zoom level for a given coordinate to get correct results. So it is often necessary to convert various coordinate values to another zoom level. All the filtering steps for example in our pipeline are done on a scaled down version by the factor 32 of the WSI to enhance the speed and get the results in reasonable time span.

A Rectangle object represents the bounds of a tile. It also wraps necessary functionality like rotation. The Grid class implements all the functionality to represent a grid of Rectangles and therefore possible tile locations that are laid over a ROI. Finally there is the GridManager class. It creates as many Grid objects for each ROI as it is specified in "grids_per_roi" and contains some convenience functions for e.g. visualization. It also merges overlapping ROIs. To check out the full spectrum of functionality of these classes checkout the code on github: https://github.com/FAU-DLM/wsi_processing_pipeline/blob/master/tile_extraction/tiles.py#L78

If "grids_per_roi" is greater than one, multiple slightly shifted grids are laid over each ROI. This increases the number of tiles and therefore the amount of training data.

This of course means, that the same tissue is present in multiple tiles, but nonetheless all tiles are unique. If "optimize_grid_angles" is true, the grid is rotated in an iterative approach by "angle_stepsize" in each iteration and the angle which results in the most tiles per ROI will be used for further calculations. This is done for each ROI individually. So the smaller "angle_stepsize" is, the closer the angle gets to the optimum, but the longer the process takes. The last important parameter in this context is "minimal_tile_roi_intersection_ratio". If it is 1.0, only tiles which lay 100% in the ROI will be considered for further processing. The closer it gets to 0.0, the more tiles can be outside of the ROI, but never completely, since 0.0 is outside of the possible range of this value.

2.5 *Tile filtering*

Among these tiles there might still be some, which are not worth to be kept. If ROIs are specified this amount should be fairly small, but if no ROIs are specified, there should be plenty to be filtered out. The user of the library can specify a tile scoring function that only takes the tile in form of a PIL image as parameter and returns a score for it. The user also has to provide a threshold for that score. All tiles with a score above this threshold pass filtering and will be considered for training.

The library provides a default tile scoring functionality which works for H&E stained slides.

$$score = 1 - \frac{10}{10 + \frac{tissuePercentage * colorFactor * saturationAndValueFactor}{1000}}$$

The scoring formula generates good results for the images in the dataset and was developed through experimentation with the training dataset.

The first criteria is the amount of tissue in a tile. To separate tissue from background we applied four filters to a tile image (Supplement 9). First the image was converted to greyscale and then its complement was created. After that Otsu's threshold was applied. Thresholding using Otsu's method is a popular thresholding technique. This technique was used in the image processing described in A Unified Framework for Tumor Proliferation Score Prediction in Breast Histopathology [18]. Otsu's method produces a binary mask in which 0 stands for background and 1 for tissue. Finally this mask is applied to the original image and background becomes black. By dividing the number of 1s by the total number of pixels the tissue percentage can be obtained.

The *colorFactor* value is used to weigh hematoxylin staining heavier than eosin staining. Utilizing the Hue-Saturation-Value (HSV) color model, broad saturation and value distributions are given more weight by the *saturationAndValueFactor*. The *score* is scaled to a value from 0.0 to 1.0.

Tissue with hematoxylin staining is most likely preferable to eosin staining. Hematoxylin stains acidic structures such as DNA and RNA with a purple tone, while eosin stains basic structures such as cytoplasm proteins with a pink tone.

Differentiating purplish shades from pinkish shades can be difficult using the RGB color space (see https://en.wikipedia.org/wiki/RGB_color_space). Therefore, to compute our $colorFactor$ value, we first convert our tile in RGB color space to HSV color space (see https://en.wikipedia.org/wiki/HSL_and_HSV). In this color model, the hue is represented as a degree value on a circle. Purple has a hue of 270 degrees and pink has a hue of 330 degrees. We remove all hues less than 260 and greater than 340. Next, we compute the deviation from purple (270) and the deviation from pink (330). We compute an average factor which is the squared difference of 340 and the hue average. Saturation and value standard deviations should be relatively broad if the tile contains significant tissue. The $colorFactor$ is computed as the pink deviation times the average factor divided by the purple deviation. It favors purple (hematoxylin stained) tissue over pink (eosin stained) tissue.

The information about one tile is then stored in a Tile object.

The result of the filtering process is a TileSummary object for each WSI. A TileSummary object contains the information about the WSI including dimensions, scaled dimensions, which were used for faster tile calculations, ROIs, the GridManager object and all tiles. It also implements some visualization methods to display the WSI with ROI and tile boundaries.

In the next step the PatientManager class in wsi_processing_pipeline.shared.patient_manager.py comes into play. Its main purpose is to manage the typically hierarchical structure of a pathological dataset. A tile belongs to a ROI. A ROI belongs to a WSI. A WSI belongs to a case and a case belongs to a patient. It is good practice to split datasets on patient level. To measure performance of a model after training usually it is not only of interest how a model performs on tile level, but also how it performs on WSI or case level. Therefore it is important to store those relationships. It is also responsible for setting the labels of each tile. The PatientManager class implements some convenience functions for dataset splitting into a training, validation and test set and for a k-fold cross validation split. It can print out a class distribution and is capable of undersampling the dataset,

In the next step the fastai [19] library takes over for training the neural network. During tile filtering the user of our library can specify in the WsiToTiles function, if each tile should be extracted and stored to disc as a PNG file. We wrote a custom fastai ImageBlock called TileImageBlock that works with fastai's data block API. This allows to renounce saving each tile to disc because the TileImageBlock can extract a tile image on the fly during the training process given the spatial information about a tile which is stored in each Tile object. This has the advantage of consuming less storage space and since it is usually necessary to play around with the parameters that are used for filtering until only the desired tiles are left, not saving the tiles is a huge speedup for this part of the process.

Our preferred library for training a neural network is fastai [19], which is built on top of Facebook's increasingly popular PyTorch [20] library.

After training has finished one wants to know how good the trained model performs on the validation or an unseen test set. For this use case we implemented the Predictor class which resides in wsi_processing_pipeline.postprocessing.predictor.py. It takes a fastai [19] Learner and one of our library's PatientManager class objects. In a first step it calculates predictions for each tile image in desired dataset. In a second step it calculates the predictions for each WSI or case by calculating the mean raw prediction for all classes for each tile and applying a threshold that can be specified for each class by the user of the library.

The last step is to evaluate the performance of the model. We therefore implemented the Evaluator class in wsi_processing_pipeline.postprocessing.evaluator.py.

Its constructor takes an instance of the above mentioned Predictor class as the only argument. It implements a few commonly used methods to measure model performance. It can calculate the per class accuracy and plot receiver operating characteristic (ROC) curves, precision recall curves, confusion matrices (Figure 1) and probability histograms (Figure 2). It can also print out sklearn's classification report and print a list of tiles with the highest losses or a list of cases, WSI or tiles sorted by a user specified metric calculated with the predictions. It is also capable of creating Gradient-weighted Class Activation Mappings (Grad-CAMs) [22].

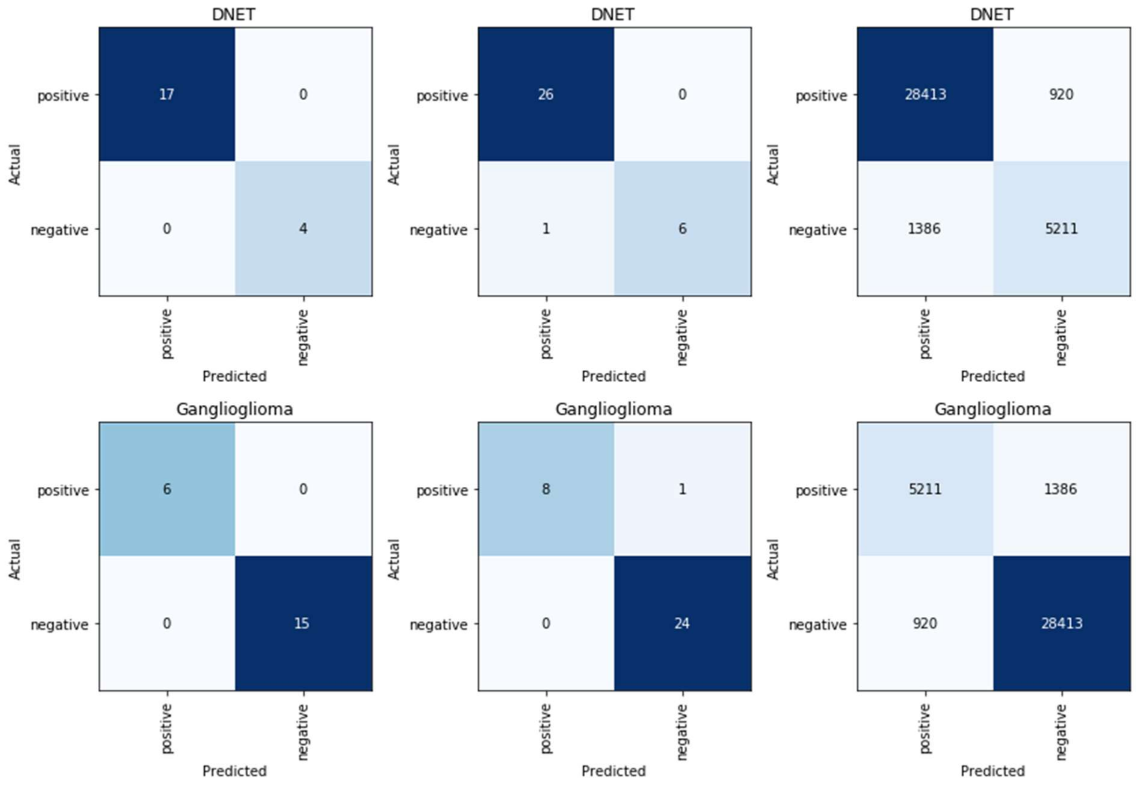

Figure 1. Results Confusion Matrices from left to right: case level, slide level, tile level.

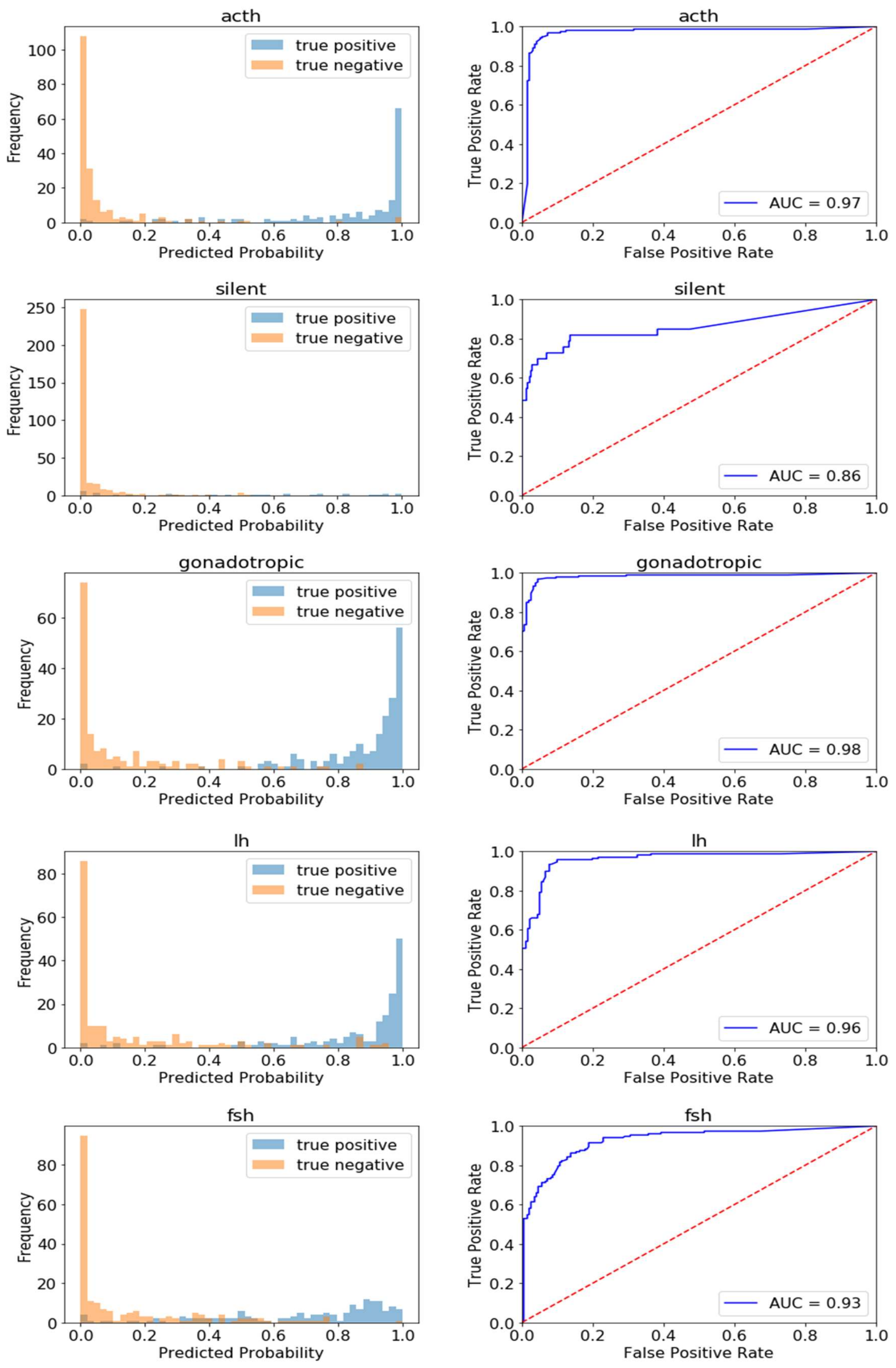

Figure 2. Results | Histograms and ROC-Curves were calculated on a case basis. The predictions were made for all 5 validation sets with the respectively corresponding model that was not trained on that validation set. So, the graphs represent the complete dataset.

2.6. Dataset preparation for both use-cases

Histopathology slides from all patients of interest for the study design were retrieved from the archives of the Dept. of Neuropathology (see below) and subsequently digitized using a Hamamatsu S60 scanner with a 40x magnification. We included only H&E stainings, thus eliminating the need for more complex and expensive immunostainings. The WSI of our dataset were reviewed by two expert neuropathologists of our institute.

Use case 1: For the DNET and ganglioglioma classifier slides from 219 patients were used. 52 of them were DNETs and 167 were ganglioglioma. Two of our expert neuropathologists in epilepsy pathology used QuPath to define polygonal ROIs containing tumor tissue in the WSI and exported their coordinates to json files. These json files were then used by the library to only extract tiles from relevant regions of the WSI. In total 171,514 tiles from GG and 34,520 tiles from DNETs with a size of 1024x1024 pixels were defined for further processing and training.

Use case 2: To train and evaluate the pituitary adenoma classifier, H&E and immunohistochemically stained (ACTH, LH, FSH) tissue slides of 410 patients were collected. 181 of these were diagnosed with corticotroph and 229 with gonadotroph adenoma of the pituitary gland (Supplement 1 and 2). Overall, the dataset consisted of 431 H&E (202 corticotroph and 229 gonadotroph) slides with the corresponding ACTH respectively LH/FSH whole slide images for comparing and identifying the correct ROI (Figure 3). The ROIs on an individual H&E slide were defined as regions, where the immunostainings showed tumor expressions of the specific hormone. Care was taken that no normal pituitary gland tissue was included (Figure 3). This time-consuming ROI selection process was necessary to ensure the correct labeling of each tile and therefore the validity of the resulting models. Otherwise, biases through wrong labeled areas could have worsened the performance. For example, areas with only connective tissue had been excluded. Also, the hormone expression of the adenoma is not homogeneously spread over the sample. This was particularly important to consider for the gonadotropic adenomas. When an adenoma expresses LH and FSH that does not mean that all subregions express both two hormones. So, there can be tiles that only get labeled with LH or FSH although the whole tumor expresses both. ROIs were defined at 40x magnification level and cropped into smaller tiles of 1024x1024 pixels to further preprocess and feed into our model (Figure 3). The tile extraction resulted in 206,517 gonadotropic and 63,893 corticotropic tiles.

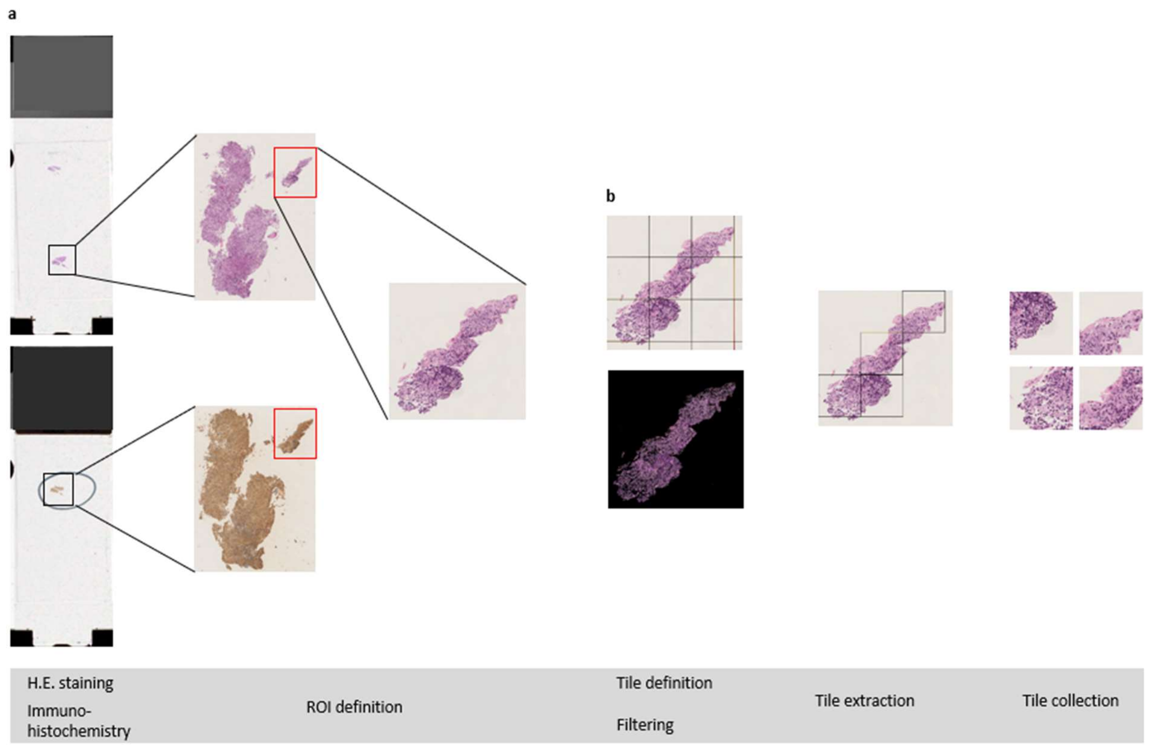

Figure 3. Tile Extraction a: We compared H&E and immuno-stained slides and extracted only those corresponding parts of the H&E stained WSI with QuPath where the immuno-stained WSI showed the expression of the hormone. b: We subdivided the image into 1024x1024 pixel tiles and used complement filter and otsu thresholds to identify tissue and background. Then we only extracted and saved those tiles that pass a scoring function that takes tissue percentage and color characteristics into account.

2.7. Convolutional Neural Network Architecture

Use case 1: For the DNET-GG classifier a ResNet50 was implemented, using the open source Python library fastai [19] which is based on PyTorch [20]. It was pretrained on ImageNet [23,24] and the classification head was replaced to predict two (DNET or GG) instead of the 1000 classes included in the ImageNet dataset. (Supplement 3).

Use case 2: For the pituitary gland classifier a ResNeXt-101-32x8d CNN architecture also pre-trained on ImageNet [23,24] was implemented. ResNeXt-101-32x8d [25,26] was chosen, as it yielded the best results with the least overfitting out of a couple of state-of-the-art network architectures including ResNet50, se_ResNeXt101_32x4d, xception and inceptionv4 (Supplement 5). The basic network architecture was not changed. Only a customized classification head (Figure 4, Supplement 3) was used to predict four instead of the 1000 ImageNet classes. It consisted of several pooling, batch normalization, dropout and fully connected layers with four final output channels with a sigmoid-activation function with a threshold of 0.5 to produce individual output probabilities representing the four classes of corticotropic adenoma, silent corticotropic adenoma, gonadotropic adenoma with the expression of LH and gonadotropic adenoma with the expression of FSH (Figure 4).

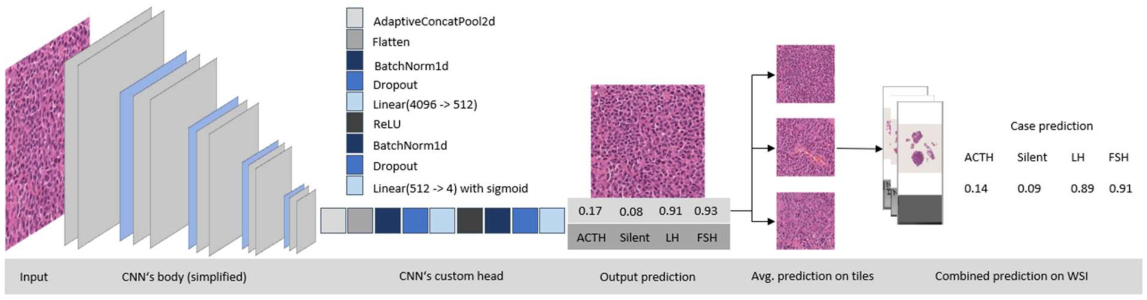

Figure 4. Prediction Pipeline A tile is forwarded through the model and the model outputs four independent probabilities for each class. If the probability is over a certain threshold (0.5) the tile gets the label. All tiles of one case are evaluated and if more than 50% of the tiles are labeled with one class, the case is also labeled with that class (majority voting).

2.8. Preprocessing and Data Augmentation

Image preprocessing is an important step in every computer vision task to augment the number of samples, to prevent overfitting, and to support the model against invariant aspects that are not correlated with the label [27,28]. First the tiles were resized to 512x512 pixel images to increase the possible batch size. Following this approach, we made sure to have a wider field of view per tile instead of the maximum possible resolution. In our approach, we used a pipeline of several augmentation techniques performed batch-wise on the GPU consisting of a random crop with reflection padding, randomly flipping (horizontal or vertical) and rotating by a multiple of 90 degrees, a random symmetric warp with a magnitude between -0.2 and 0.2, a random rotation between -10 and +10 degrees, a random zoom with a zoom factor between 1.0 and 1.1, a random change in brightness with a factor between 0.4 and 0.6 where a factor of 0 will transform the image to black, a factor of 1 will transform the image to white and a factor of 0.5 doesn't adjust the brightness. Furthermore, an augmentation on contrast of the image was applied with a factor between 0.8 and 1.25 where a factor of 0 will transform the image to grey, a factor over 1 will transform the picture to super-contrast and a factor = 1 does not adjust the contrast. These augmented images were then normalized. The augmentations were applied on the fly with a randomness factor for reproducibility for every batch so that there was no need to save augmented images and one image could be augmented in multiple ways. This whole approach ensures that out of one image multiple new images of the same class can be obtained multiplying the number of images available for training the neural network.

2.9. Training and Evaluation

Training was performed with 16-bit precision floating-point numbers [29] using the Adam-Optimizer [30] and the initial learning rate was determined by using fastai's learning rate finder (Supplement 4). Learning rate was adjusted during the training according to the one-cycle-policy [31]. The batch size was twelve for the pituitary adenoma classifier and 35 for the DNET-GG classifier. At first, only the randomly initialized custom head (Figure 2, Supplement 3) was trained for five epochs with a maximum learning rate of 10^{-3} (Supplement 4) in both projects to not interfere with the pretrained weights of the CNN's body. Thereafter the body's layers were unfrozen, and the complete network was trained for ten epochs with differential learning rates between 10^{-9} and 10^{-6} for the pituitary gland adenoma classifier and between 10^{-8} and 10^{-6} for the DNET-GG classifier (Supplement 4) where earlier layers are trained with a lower learning rate than the later ones. The idea behind this is, to maintain the basic image-classification patterns of the pretrained model and prevent overfitting. Training performance was controlled using accuracy with a threshold of 0.5 as metric per tile and the used loss function was binary cross-entropy loss. Model parameters were saved every epoch and the weights of the epoch with the best results were used for evaluation. We further evaluated model performance with 5-fold cross-validation, without having any training- and validation-slide and patient overlap. After the training, predictions on the five validation sets were calculated with the corresponding model based on the combined predictions of all tiles of a case. The prediction for a case was calculated using majority voting for the pituitary gland adenoma classifier and the arithmetic mean of the raw predictions (between 0.0 and 1.0) of all the case's tiles for the DNET-GG classifier. These results were then combined and used to calculate true and false-positive rates, which were then used to plot Receiver Operating Characteristic curves, true/false positive frequency histograms, and in conjunction with false-negative rates to plot precision-recall curves.

Since silent corticotroph adenomas only made up 9.7% of the dataset, we decided to train a second neural net on an undersampled training set. The original training set (80% of the complete dataset) consisted of 226,422 tiles from which 59% were positive for LH, 62% for FSH, 22% for ACTH, and 9.4 % were silent corticotroph adenomas. After the downsampling procedure, 54,713 tiles were left from which 43% were positive for LH, 43% for FSH, 43% for ACTH and 39 % were silent corticotroph adenomas. We assured that at least 30 tiles per WSI were left after downsampling. Again, we used the resnext101_32x8d architecture. The head was trained for five epochs with a maximum learning rate of 10^{-3}. The complete model was then trained for ten epochs with maximum discriminative learning rates ranging from 10^{-7} to 10^{-5}. In both cases, the one-cycle learning rate policy was used with minimum learning rates of 1/25 of the maximum learning rates.

2.10. Hardware

We implemented our approach on a local server running Ubuntu (18.04 LTS) with one NVIDIA GeForce GTX 1080Ti and one NVIDIA Titan XP, AMD CPU (AMD Ryzen Threadripper 1950X 16 × 3.40 GHz), 128 Gb RAM, CUDA 10.2, and cuDNN 7.

2.11. Availability and implementation

The datasets generated and analyzed during the presented study are not publicly available, but parts of the pipeline used in this project including training and visualization are available on our Project Homepage.

https://github.com/FAU-DLM/wsi_processing_pipeline
https://github.com/ChristophNeuner/pituitary_gland_adenomas
https://github.com/ChristophNeuner/DNET_vs_Ganglioglioma

3. Results

3.1. Use case 1: *DNET- GG classifier*

We evaluated the performance on the validation set which made up 20% of the whole dataset and which was not used for training. It consisted of 24 slides of ganglioglioma and seven slides of DNET. 29,333 tiles were extracted from the GG slides and 6,597 tiles were extracted from the DNET slides for evaluation. No hyperparameter tweaking was performed, which could have led to overfitting on the validation set. On tile level the accuracy was 0.936 and on slide level 0.968. The AUC on tile level was 0.93 and 1.00 on slide level for the ROC-curve. The average precision calculated from precision and recall was 0.88 for DNET and 0.97 for GG on tile level. On slide level it was 1.00 for DNET and GG. (Figures 5 and 6)

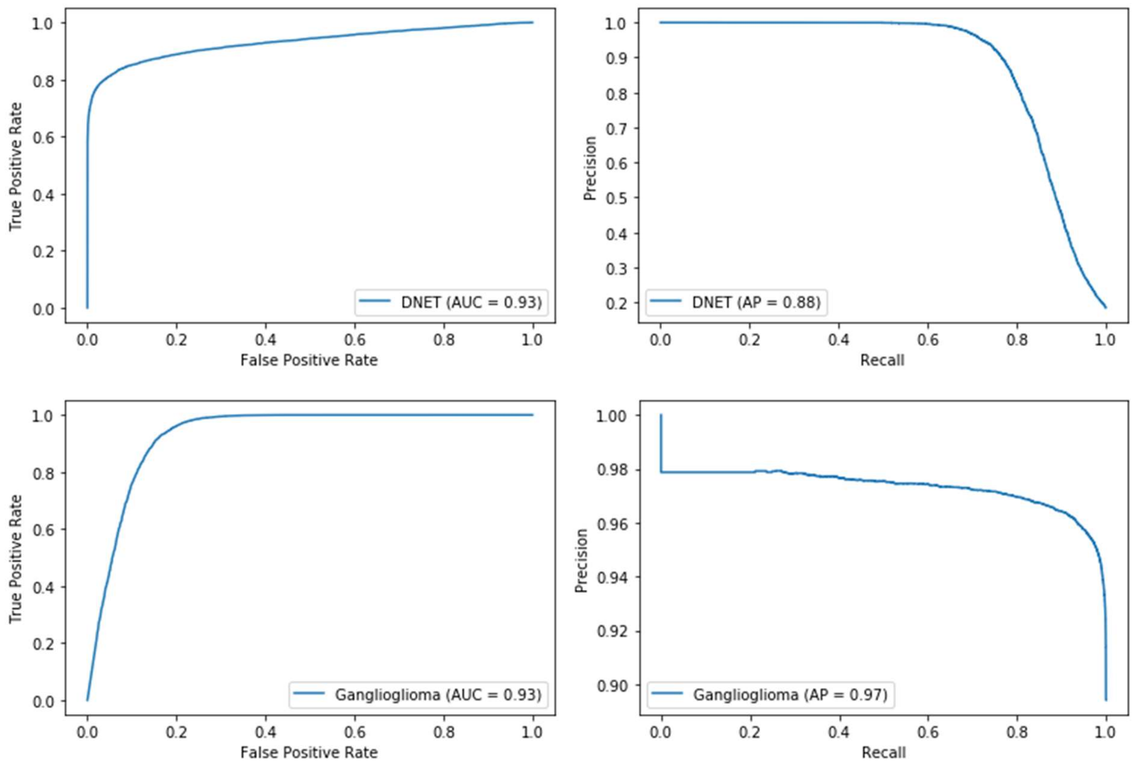

Figure 5. Results | ROC (left) and precision recall curves (right) on tile level.

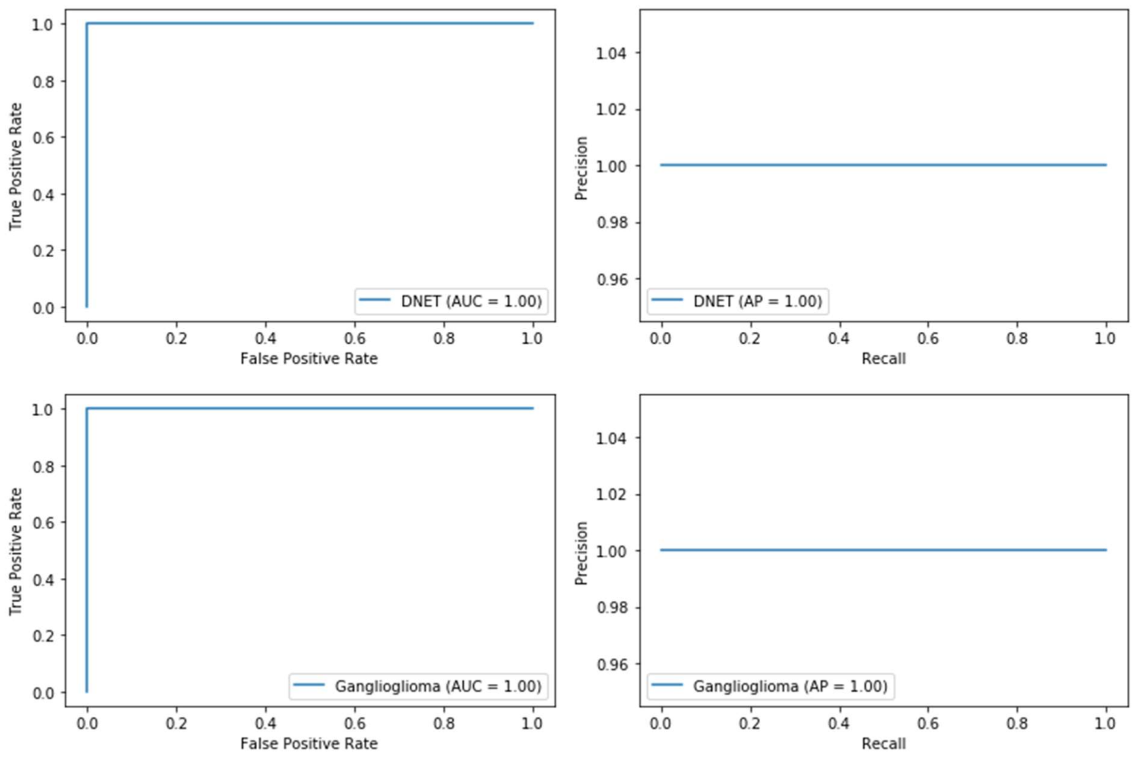

Figure 6. Results ǀ ROC (left) and precision recall curves (right) on slide level.

3.2. Use case 2: *Pituitary adenoma classifier*

All CNN were trained to classify the ROIs containing adenoma and surrounding tissue. First, we performed a study to determine which model to use for our classification task. We tested ResNet50, ResNet101, ResNet152, DenseNet121, Xception, Inceptionv4, se_ResNext101_32x4d and ResNext101_32x8d. We compared those models on a predefined validation set with accuracy calculated on a case basis for each class with a threshold of 0.5 (Supplement 5). Inceptionv4, se_ResNext101_32x4d and ResNext101_32x8d showed similar promising results. We decided upon ResNext101_32x8d because of the slightly better test-set results. During training validation accuracies mostly stayed above training accuracies and validation loss stayed below training loss values, indicating little to no overfitting on the training dataset. We finally evaluated our model via 5-fold cross-validation. For each model within the process of cross-validation, we took 80% of the dataset as training data and 20% as validation data. There was no overlap between these five validation sets. All five validation sets showed similar AUCs with no significant outliers (Supplement 6). After that predictions were made for all tiles of the five validation sets with the respectively corresponding model that was not trained on that particular validation set. Via majority voting with a threshold of 0.5 we then calculated the labels on a case basis and computed AUCs of ROC curves for each class. If more than 50% of the tiles of one case were labeled with the class ACTH, the whole case got the label ACTH.

For ACTH the AUC of the ROC curve was 0.97 with a proportion of 44.7% of all cases. The AUC for silent ACTH was 0.86 with a proportion of 9.7%. The AUC for gonadotropic (LH and/or FSH) was 0.98 with a proportion of 55.3%. The AUCs of LH and FSH alone were 0.96 and 0.93 with proportions of 48.1% and 43.8% (Figure 2). Since the silent ACTH cases only made up 9.7% of the dataset the AUC of 86% of the ROC curve could have simply been a result of guessing. Therefore, we also calculated a precision-recall curve (Figure 7) which resulted in an AUC of 0.71 and furthermore trained another neural net on an undersampled dataset as described in the last paragraph of “Training and Evaluation”. We reached an accuracy of 88.6% and an AUC of 0.83 for the ROC-curve on the validation set for the silent ACTH class (Figure 8).

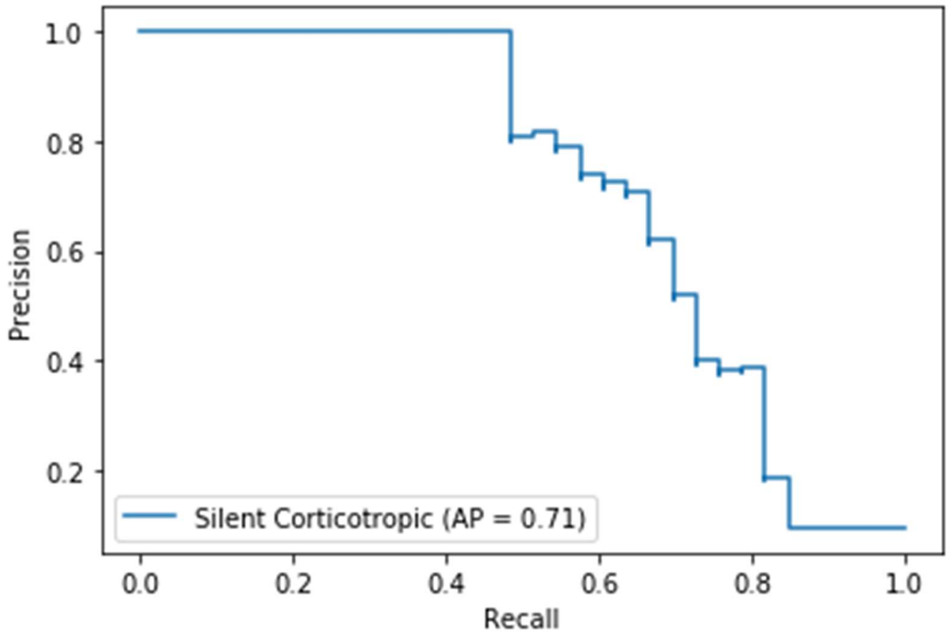

Figure 7. Results | Precision-recall curve for the class silent corticotropic adenoma of the models from the 5-fold cross-validation which were trained on the unevenly distributed training set, in which silent-corticotroph adenoma made up only 9.7% of the tiles.

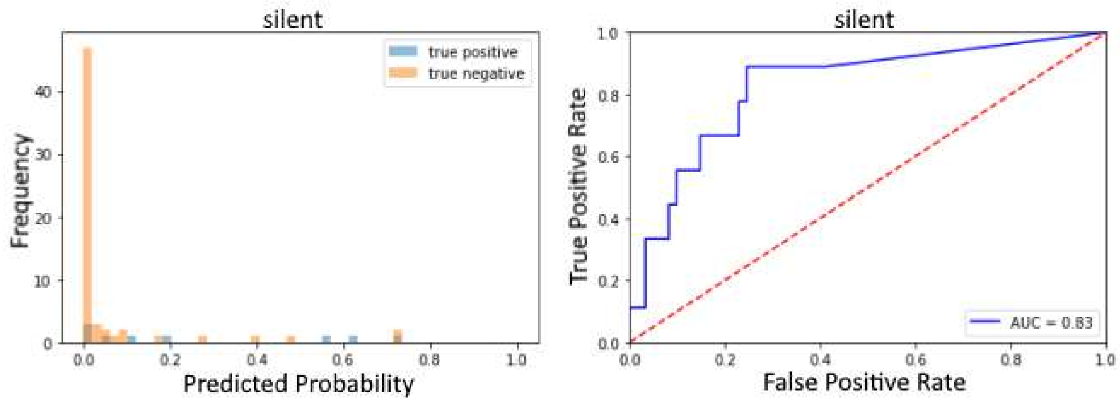

Figure 8. Results | Probability Histogram and ROC-Curve for the class silent corticotroph adenoma of the model that was trained on an undersampled training set, in which all four classes were evenly distributed.

4. Discussion

We developed a whole slide image processing library [15] addressing the needs of researchers to assess different DL tasks without the hurdles of complex dataset management. The large size of WSI and annotation of multiple regions of interest tend to increase such technical obstacles. It is also desirable to extract all tiles on-the-fly during training and only save their spacial information but not the images. This pipeline has the advantage of being more flexible. It is not necessary anymore to repeatedly store extracted tiles as images to disc, saving space and time. Also the evaluation of the trained model requires more steps when dealing with WSI. Results on the tile level are only of limited significance. They have to be transformed into predictions for the complete WSI and the entire case. For histopathologists or expert clinicians addressing a clinical hypothesis, these hurdles may become a real burden. Also DL experts familiar with the usage of DL frameworks may underestimate the specific handling of digital pathology-associated tasks. The new library provides convenient ways of dealing with WSI in the realm of Neuropathology thereby facilitating access to DL for both groups of researchers.

Access from and to different levels of magnification, region of interest definition, and handling as well as dataset splitting are essential mechanisms and tend to be technically intricate. The library manages these crucial steps and offers default parameters enabling the user to focus on the problem-specific tasks. For the specific use-cases addressed in this study, the library facilitated the management of pre-extracted image patches for a given patient as well as extraction of image patches on-the-fly from predefined ROI. Our evaluation of different state-of-the-art model architectures to identify the most suitable model for the problem-specific tasks, i.e. best classification results and least overfitting, resulted in the selection of resnet50 for the first use-case and the resnext101_32x8d [25,27] architecture for the second use-case. We believe that these rather big networks with lots of parameters worked well, because of their big input image size of 512x512 pixels. On smaller images networks with less parameters tend to work better in our experience [3]. A crucial step in our pipeline was among sufficient training data the way of image preprocessing. One part of this aspect was image augmentation to increase variance presented to the network [32]. Normalization of the input data was done with the mean and standard deviation of our own dataset. Fastai [19] does this conveniently for the user.

Use-case 1: In the first use case we developed a DL approach to distinguish between two epilepsy associated tumors, the GG and the DNET. Since unlike DNET some GG can undergo malignant transformation [8,9], a precise distinction between these two entities is crucial. We were able to demonstrate that a CNN can differentiate these two entities with a very high accuracy only using H&E-stained slides. This confirms the potential of DL in assisting pathologists in their decision making diagnostic process and to eventually reduce the necessity for further stains.

Use-case 2: In the second use case addressed we developed a DL approach to help to diagnose the entity of pituitary adenomas without the necessity of additional immunohistochemical stainings. Additionally, we could show that even a clinical parameter, the clinical occurrence of M. Cushing of corticotroph adenomas, might be hidden within the tissue as it could successfully be recognized by our neural network approach. This evidence supports the hypothesis that clinical parameters can be found within histomorphology and that distinct features may be revealed by DL in terms of imaging biomarkers. Guided Grad-CAMs [22] could now be used to visualize the decision making and to teach pathologists which morphological structures are crucial for the network in its decision making process.

We addressed the classification task on predictions per tile and collected all votes for the given slides of a patient's case. We then obtained the final diagnosis by majority voting to get predictions on a case basis. If more than 50% of the tiles of one case were labeled with one class, the case was given that class label. We chose that option for two reasons.

First, different from finding metastasis in lymph nodes where high sensitivity is needed, histological slides from pituitary adenomas usually contain massive adenoma, hence most of the tissue on the slide belongs to the tumor. Second, time was not a major concern. We could simply take and analyze all possible tiles instead of only taking a representative batch for inference.

Limitations and potential solutions moving into the future

A well-recognized obstacle in digital pathology represents batch effects including variation in staining intensity or fixation artifacts [4,33]. We contained such batch effects in our input data through hand-picked ROI and normalization. We did not directly address the problem of stain normalization [34] for this dataset, because all staining was performed in a single lab and only one device was used for scanning. For further usage of our model in a production environment with whole slide images from other institutes, this would be crucial. We are continuously working on this issue to make our models more robust in the future.

Histopathology analysis represents a gold-standard in tumor diagnosis as it often directs further treatment. Adenomas of the pituitary gland, although routinely classified by immunohistochemical profiling of their neuroendocrine axis, are in urgent need of a clinically meaningful histopathology classification of their risk for relapse. This was partially addressed by the WHO classification from 2004 and 2016. The criteria of atypia to label more aggressive adenomas have been removed, however, as it has missed to proof as predictive marker [14,35]. The "silent" corticotroph class of our dataset did represent another clinical parameter of interest and was remarkably well recognized by our network, even in the evenly distributed dataset. The good classification result of the "silent" corticotroph class in our study shows that neuronal networks are capable in revealing such clinical information hidden within tissue slides and hence it may also be possible to extract a clinical relapse parameter from tissue slides via DL.

In conclusion, we developed a convenient open-access library compatible with fastai to support hypothesis driven DL research projects in the realm of neuropathology. Both use-cases demonstrated the successful diagnosis of adenoma of the pituitary gland and distinguishing between DNET and GG by H&E-stained slides only and without the necessity of cost- and labor-intense immunohistochemistry staining.

Supplementary Materials: The following are available online at

www.mdpi.com/xxx/s1, Table S1: Dataset

www.mdpi.com/xxx/s2, Table S2: Class Distribution

www.mdpi.com/xxx/s3, Figure S3: Custom head (Pytorch)

www.mdpi.com/xxx/s4, Figure S4: Learning rate finder pituitary adenoma classifier

www.mdpi.com/xxx/s5, Table S5: Evaluated Networks

www.mdpi.com/xxx/s6, Table S6: AUCs of the ROC-curves for the five validation sets of 5-fold cross-validation

www.mdpi.com/xxx/s7, Figure S7: QuPath

www.mdpi.com/xxx/s8, Figure S8: ROIs with overlaid grids

www.mdpi.com/xxx/s8, Figure S9: Tissue filtering

Author Contributions: "Conceptualization, Christoph Neuner, Samir Jabari, Roland Coras and Ingmar Blümcke; methodology, Christoph Neuner and Samir Jabari; software, Christoph Neuner; validation, Christoph Neuner, Samir Jabari, Roland Coras and Ingmar Blümcke; formal analysis, Christoph Neuner and Samir Jabari; investigation, Christoph Neuner and Samir Jabari; resources, Samir Jabari, Ingmar Blümcke, Sven-Martin Schlaffer and Michael Buchfelder; data curation, Christoph Neuner and Alexander Popp; writing—original draft preparation, Christoph Neuner; writing—review and editing, Samir Jabari, Roland Coras, Ingmar Blümcke, Sven-Martin Schlaffer and Michael Buchfelder; visualization, Christoph Neuner; supervision, Samir Jabari; project administration, Samir Jabari; funding acquisition, Samir Jabari and Ingmar Blümcke. All authors have read and agreed to the published version of the manuscript."

Funding: Please add: "This research was funded by the Interdisciplinary Center for Clinical Research (IZKF) at the University Hospital of the University of Erlangen-Nuremberg , grant number Junior Project J81.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.
Data Availability Statement: he code used here is available from the following three GitHub repositories: https://github.com/FAU-DLM/wsi_processing_pipeline,
https://github.com/ChristophNeuner/pituitary_gland_adenomas,
https://github.com/ChristophNeuner/DNET_vs_Ganglioglioma

The whole-slide images used here are not publicly available.

Acknowledgments: The present work was performed in fulfillment of the requirements of the Friedrich-Alexander Universität Erlangen-Nürnberg (FAU) for obtaining the degree 'Dr. med.' of Christoph Neuner. The work was supported by the Interdisciplinary Center for Clinical Research (IZKF) at the University Hospital of the University of Erlangen-Nuremberg (Junior Project "J81"). We would also like to thank NVIDIA for the donation of a Titan XP.

Conflicts of Interest: The authors declare no conflict of interest

References

1. Ehteshami Bejnordi B, Veta M, Johannes van Diest P, van Ginneken B, Karssemeijer N, Litjens G, et al. Diagnostic Assessment of Deep Learning Algorithms for Detection of Lymph Node Metastases in Women With Breast Cancer. JAMA [Internet]. 2017/12/14. 2017;318(22):2199–210. Available from: https://www.ncbi.nlm.nih.gov/pubmed/29234806
2. Arvaniti E, Fricker KS, Moret M, Rupp N, Hermanns T, Fankhauser C, et al. Automated Gleason grading of prostate cancer tissue microarrays via deep learning. Sci Rep [Internet]. 2018/08/15. 2018;8(1):12054. Available from: https://www.ncbi.nlm.nih.gov/pubmed/30104757
3. Kubach J, Muhlebner-Fahrngruber A, Soylemezoglu F, Miyata H, Niehusmann P, Honavar M, et al. Same same but different: A Web-based deep learning application revealed classifying features for the histopathologic distinction of cortical malformations. Epilepsia [Internet]. 2020;61(3):421–32. Available from: https://onlinelibrary.wiley.com/doi/abs/10.1111/epi.16447
4. Tang Z, Chuang K V, DeCarli C, Jin L-W, Beckett L, Keiser MJ, et al. Interpretable classification of Alzheimer's disease pathologies with a convolutional neural network pipeline. Nat Commun [Internet]. 2019;10(1):2173. Available from: https://www.ncbi.nlm.nih.gov/pubmed/31092819
5. Janowczyk A, Madabhushi A. Deep learning for digital pathology image analysis: A comprehensive tutorial with selected use cases. J Pathol Inf [Internet]. 2016/08/27. 2016;7:29. Available from: https://www.ncbi.nlm.nih.gov/pubmed/27563488
6. Coudray N, Ocampo PS, Sakellaropoulos T, Narula N, Snuderl M, Fenyo D, et al. Classification and mutation prediction from non-small cell lung cancer histopathology images using deep learning. Nat Med [Internet]. 2018/09/19. 2018;24(10):1559–67. Available from: http://www.ncbi.nlm.nih.gov/pubmed/30224757
7. Blümcke I, Coras R, Wefers AK, Capper D, Aronica E, Becker A, et al. Review: Challenges in the histopathological classification of ganglioglioma and DNT: microscopic agreement studies and a preliminary genotype-phenotype analysis. Neuropathol Appl Neurobiol [Internet]. 2019;45(2):95–107. Available from: https://onlinelibrary.wiley.com/doi/abs/10.1111/nan.12522
8. Majores M, von Lehe M, Fassunke J, Schramm J, Becker A, Simon M. Tumor Recurrence and Malignant Progression of Gangliogliomas. Cancer. 2008;113:3355–63.
9. Selvanathan S, Hammouche S, Salminen H, Jenkinson M. Outcome and prognostic features in anaplastic ganglioglioma: Analysis of cases from the SEER database. J Neurooncol. 2011;105:539–45.
10. Thom M, Toma A, An S, Martinian L, Hadjivassiliou G, Ratilal B, et al. One Hundred and One Dysembryoplastic Neuroepithelial Tumors: An Adult Epilepsy Series With Immunohistochemical, Molecular Genetic, and Clinical Correlations and a Review of the Literature. J Neuropathol Exp Neurol [Internet]. 2011;70(10):859–78. Available from: https://doi.org/10.1097/NEN.0b013e3182302475
11. Slegers RJ, Blumcke I. Low-grade developmental and epilepsy associated brain tumors: a critical update 2020. Acta Neuropathol Commun [Internet]. 2020;8(1):27. Available from: https://doi.org/10.1186/s40478-020-00904-x
12. Ezzat S, Asa SL, Couldwell WT, Barr CE, Dodge WE, Vance ML, et al. The prevalence of pituitary adenomas. Cancer [Internet]. 2004;101(3):613–9. Available from: https://acsjournals.onlinelibrary.wiley.com/doi/abs/10.1002/cncr.20412
13. Aflorei ED, Korbonits M. Epidemiology and etiopathogenesis of pituitary adenomas. J Neurooncol [Internet]. 2014;117(3):379–94. Available from: https://doi.org/10.1007/s11060-013-1354-5
14. Inoshita N, Nishioka H. The 2017 WHO classification of pituitary adenoma: overview and comments. Brain Tumor Pathol [Internet]. 2018;35(2):51–6. Available from: https://doi.org/10.1007/s10014-018-0314-3
15. Neuner C. python-wsi-preprocessing [Internet]. https://github.com/FAU-DLM/python-wsi-preprocessing. GitHub; 2019. Available from: https://github.com/FAU-DLM/python-wsi-preprocessing
16. Veta M, Heng YJ, Stathonikos N, Bejnordi BE, Beca F, Wollmann T, et al. Predicting breast tumor proliferation from whole-slide images: The TUPAC16 challenge. Med Image Anal. 2019 May 1;54:111–21.
17. Eriksson D. python-wsi-preprocessing [Internet]. https://github.com/deroneriksson/python-wsi-preprocessing. GitHub; 2018. Available from: https://github.com/deroneriksson/python-wsi-preprocessing
18. Paeng K, Hwang S, Park S, Kim M. A Unified Framework for Tumor Proliferation Score Prediction in Breast Histopathology. In: DLMIA/ML-CDS@MICCAI. 2017.
19. Howard J, Gugger S. Fastai: A Layered API for Deep Learning. Information [Internet]. 2020;11(2):108. Available from: https://www.mdpi.com/2078-2489/11/2/108
20. Paszke A, Gross S, Massa F, Lerer A, Bradbury J, Chanan G, et al. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In: Wallach H, Larochelle H, Beygelzimer A, d\textquotesingle Alché-Buc F, Fox E, Garnett R, editors. Advances in Neural Information Processing Systems [Internet]. Curran Associates, Inc.; 2019. Available from: https://proceedings.neurips.cc/paper/2019/file/bdbca288fee7f92f2bfa9f7012727740-Paper.pdf
21. Sezgin M, Sankur B. Survey over image thresholding techniques and quantitative performance evaluation. J Electron Imaging [Internet]. 2004;13(1). Available from: https://doi.org/10.1117/1.1631315
22. Selvaraju RR, Das A, Vedantam R, Cogswell M, Parikh D, Batra D. Grad-CAM: Why did you say that? [Internet]. arXiv e-prints. 2016. Available from: https://ui.adsabs.harvard.edu/abs/2016arXiv161107450S
23. Simonyan K, Zisserman A. Very Deep Convolutional Networks for Large-Scale Image Recognition [Internet]. arXiv e-prints. 2014. Available from: https://ui.adsabs.harvard.edu/abs/2014arXiv1409.1556S
24. Deng J, Dong W, Socher R, Li L, Kai L, Li F-F. ImageNet: A large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition. 2009. p. 248–55.
25. Saining X, Ross G, Piotr D, Zhuowen T, He K. Aggregated Residual Transformations for Deep Neural Networks [Internet]. arXiv e-prints. 2016. Available from: https://ui.adsabs.harvard.edu/abs/2016arXiv161105431X/abstract
26. Cadene R. pretrained PyTorch models [Internet]. https://github.com/Cadene/pretrained-models.pytorch. GitHub; 2019. Available from: https://github.com/Cadene/pretrained-models.pytorch

27. Wu R, Yan S, Shan Y, Dang Q, Sun G. Deep Image: Scaling up Image Recognition [Internet]. arXiv e-prints. 2015. Available from: https://ui.adsabs.harvard.edu/abs/2015arXiv150102876W
28. Wong SC, Gatt A, Stamatescu V, McDonnell MD. Understanding data augmentation for classification: when to warp? [Internet]. arXiv e-prints. 2016. Available from: https://ui.adsabs.harvard.edu/abs/2016arXiv160908764W
29. Micikevicius P, Narang S, Alben J, Diamos G, Elsen E, Garcia D, et al. Mixed Precision Training [Internet]. 2017. p. arXiv:1710.03740. Available from: https://ui.adsabs.harvard.edu/abs/2017arXiv171003740M
30. Kingma DP, Ba J. Adam: A Method for Stochastic Optimization. arXiv e-prints. 2014 Dec;arXiv:1412.6980.
31. Smith LN. Cyclical Learning Rates for Training Neural Networks [Internet]. arXiv e-prints. 2015. Available from: https://ui.adsabs.harvard.edu/abs/2015arXiv150601186S
32. Perez L, Wang J. The Effectiveness of Data Augmentation in Image Classification using Deep Learning. 2017;
33. Madabhushi A, Lee G. Image analysis and machine learning in digital pathology: Challenges and opportunities. Med Image Anal [Internet]. 2016/07/18. 2016;33:170–5. Available from: https://www.ncbi.nlm.nih.gov/pubmed/27423409
34. Anghel A, Stanisavljevic M, Andani S, Papandreou N, Rüschoff JH, Wild P, et al. A High-Performance System for Robust Stain Normalization of Whole-Slide Images in Histopathology. Front Med [Internet]. 2019;6:193. Available from: https://pubmed.ncbi.nlm.nih.gov/31632974
35. Mete O, Lopes MB. Overview of the 2017 WHO Classification of Pituitary Tumors. Endocr Pathol [Internet]. 2017;28(3):228–43. Available from: https://doi.org/10.1007/s12022-017-9498-z