Self-Supervised Learning in Histopathology: New Perspectives for Prostate Cancer Grading.

Markus Bauer[1] and Christoph Augenstein[1]

Center for Scalable Data Analytics and Artificial Intelligence (ScaDS.AI), Dresden/Leipzig, Germany
{bauer,augenstein}@wifa.uni-leipzig.de

Abstract. The prostate carcinoma (PCa) is the second most common cause of cancer-deaths among men. To estimate the appropriate therapy pathway after diagnosis, the Gleason score (GS) has been established as an international measure. While the GS has been proven to be a good tool for tumour assessment, it naturally suffers from subjectivity. Especially for cancers of lower to medium severity, this leads to inter- and intra observer variability and a remarkable amount of over- and under therapy. The PCa thus is in the focus of various research works, that aim to improve the grading procedure. With recently emerging AI technologies, solutions have been proposed to automate the GS-based PCa-grading while keeping predictions consistent. Current solutions, however, fail to handle data variability arising from preparation differences among hospitals and typically require a large amount of annotated data, which is often not available. Thus, in this paper, we propose self-supervised learning (SSL) as a new perspective for AI-based PCa grading. Using several thousand PCa cases, we demonstrate that SSL may be a feasible alternative for analysing histopathological samples and pretraining grading models. Our SSL-pretrained models extract features related to the Gleason grades (GGs), and achieve competitive accuracy for PCa downstream classification.

Keywords: Prostate Cancer · Self-Supervised Learning · Artificial Intelligence.

1 Introduction

AI technologies are already a widespread choice in various histopathological applications, including those focusing on the PCa. Current research work and emerging commercial solutions use supervised training to create convolutional neural networks (CNNs) that can extract features that match cancer-driven morphological changes and thus can reproduce the GS.

Impressive results for the PCa can be found, even in early works such as the ones of Arvaniti et al. [1] or Nagpal et al. [19], where $\sim 70\%$ of the expert's GS could be reproduced. Bulten et al. [5] achieve an area under the receiver operating characteristics curve (AUC-ROC) of 0.984 for the tumour vs. non-tumour problem. Tolkach et al. [24] even achieve a binary accuracy of over 95% using

a slightly more complex deep learning pipeline. Similar results are achieved by Ström et al. [23] with an AUC-ROC of 0.997. Finally, using CNNs for PCa grading is also part of various international challenges such as the PANDA challenge (c.f. [3] et al.), where a maximum quadratically weighted kappa of 0.86 was achieved by the best team on international validation data. Supervised training methods, however, require huge amounts of labelled data, which is often not available, and suffer from limited annotation quality or missing pathological consensus. Hence, poor generalization is often observed, as results worsen significantly for data which originates from other laboratories (c.f. [22]).

Recent advances in computer vision have shown, that SSL may be of value, to omit these issues, and may even improve performance achieved by supervised models [8,6,13,18,29]. To do so, SSL uses CNNs and visual transformers (ViTs)[16], but generates training signals from the data itself rather than using manual annotations. Early works such as the one of Bulten et al. [4], who applied SSL to PCa data using an autoencoder [26] could show that this method may be feasible for PCa histopathological analysis. Their approach, however, fails to distinguish more groups than benign, stroma and tumour. The problem seems to arise due to the autoencoder's limited capability, to extract robust, discriminative features.

More recent works suggest the definition of pretext tasks, that do not require a generative model, as in the case of an autoencoder. Such tasks may be the prediction of an image's rotation [10], solving of jigsaw puzzles [20], predicting generated pseudo-labels [27], or comparing feature vectors in a contrastive setup using an original and augmented image version [18,29,8,13]. The value of SSL for histopathology is demonstrated by Yan et al. [28], who trained an SSL model on various publicly available histopathology datasets and almost achieved supervised performance on the CAMELYON dataset[1].

Currently, only few works have investigated the value of SSL for PCa histopathological analysis. Thus, we evaluated recent SSL methods using the PANDA dataset. We conclude that SSL may be a good future path and direction to eliminate human bias from cancer grading while keeping a high standard of assessment accuracy and reproducibility. The following sections provide information about our experimental analysis and conclusions.

2 Materials and Methods

To train histopathological PCa models using SSL, we implemented a processing pipeline as depicted in Fig. 1. First, we sample image patches from a few thousand original core needle biopsy (CNB) images and select up to 16 patches per CNB according to the most amount of relevant tissue. Additionally, multiple CNBs are filtered as they are duplicates or contain very noisy labels, as suggested in the winning solution[2] of the PANDA challenge. For each image patch X, we then create multiple augmented versions $\hat{X}$ and process them using

[1] https://camelyon17.grand-challenge.org
[2] kaggle.com/competitions/prostate-cancer-grade-assessment/discussion/169143

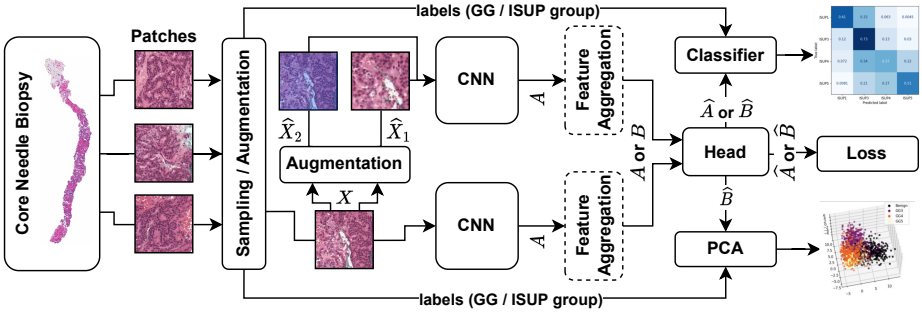


Fig. 1. Overview of the training and validation method.

a feature encoder. It's noteworthy to say that for one of our approaches (SimCLR) only augmented versions of X will be fed to the CNN. To perform whole-slide-image(WSI)-wide prediction, information from multiple patches needs to be aggregated. Thus, we implement the feature aggregation as suggested by the PANDA baseline[3] as a configurable alternative to processing the individual image patches directly. The key aspect of the feature aggregation is to concatenate multiple CNN-extracted features A of the same WSI in a global feature matrix B, such that

$$B = \text{concat}(A): \ \mathbb{R}^{(BS\cdot N)\times C\times I\times J} \rightarrow \mathbb{R}^{BS\times C\times (N\cdot I)\times J} \tag{1}$$

whereas BS is the batch size, I and J are the feature shape, C are the kernel map channels and N is a fixed number of patches. The features of the original and augmented patches are then processed by fully connected layers (projection heads) followed by pooling. Afterwards, a contrastive loss is calculated based on these projected feature vectors $\hat{A}$, or $\hat{B}$ respectively, as described in subsection 2.1. Finally, we perform multiple downstream tasks for evaluation, as described in section 3.

2.1 Self-Supervised Training Methods

The main idea of self-supervised learning (SSL) is to train a neural network in such a way that it learns to capture fundamental morphological properties of images without relying on labelled data. In traditional supervised learning, a neural network is trained on a labelled dataset, where each image is associated with a specific class or label. However, obtaining labelled data can be expensive and time-consuming, especially in domains like medical imaging.

In SSL, instead of using external labels, the network is trained to predict certain image modifications or transformations applied to the original image. For example, the network may be trained to predict the rotation angle [10] or to withstand cropping [8,6,7] applied to the image. Let's denote the unlabelled

[3] kaggle.com/competitions/prostate-cancer-grade-assessment/discussion/146855

dataset as $D = x_1, x_2, ..., x_n$, where x_i represents an input image in the dataset and n is the number of images in the dataset. The goal of SSL is to learn a representation function $\phi(x)$ that maps each input image x_i to a feature vector $\phi(x_i)$ in a latent space. The overall SSL objective can be formulated as follows:

$$\theta^* = \arg\min_{\theta} \left(\frac{1}{n} \sum_{i=1}^{n} L\left(\Phi(x_i; \theta), T(x_i)\right) \right) \tag{2}$$

whereas

- θ^* represents the optimal model parameters that minimize the overall loss.
- $\phi(x_i; \theta)$ is the feature vector obtained by applying the representation function ϕ with parameters θ to the input image x_i.
- $T(x_i)$ is the target label or artificial target for the pretext task associated with the input image x_i.
- $L(\phi(x_i; \theta), T(x_i))$ is the loss function that measures the discrepancy between the predicted target ($\phi(x_i; \theta)$) and the actual target ($T(x_i)$) for the pretext task.

The advantage of using SSL in PCa histopathology lies in its potential to extract discriminative features that are not biased by subjective factors, in contrast to directly optimising to the Gleason score. Traditionally, grading PCa involves human pathologists assigning a GS to each tissue sample based on visual appearance. However, this process can be subject to inter-observer variability and lacks objectivity. SSL circumvents the need for GS labels and allows the model to identify intrinsic morphological patterns in the images, which could directly correlate with clinical outcomes like time to biochemical recurrence (BCR) or death of disease (DoD). This opens up new avenues for computer-aided diagnosis and precision medicine in the context of prostate cancer and other medical imaging applications.

SimCLR is one of the earliest, yet still popular, SSL methods. It involves processing images X into augmented versions x_i and x_j that will be fed through a convolutional neural network (CNN) trunk, followed by a fully connected multilayer perceptron (MLP). Augmentations may be colour jitter, cropping and scaling of smaller image parts, or blurring the images, as described in section 3. This process generates feature vectors z_i and z_j of configurable length for all images.

To train the SimCLR model, an important component is the InfoNCE loss [21]. The objective of the InfoNCE loss is to maximize the similarity between feature vectors of the original image's augmented versions (z_i, z_j), while also minimizing the similarity to feature vectors of all other images' augmentations z_k in the batch. This encourages the model to learn representations that effectively capture the essential information present in the original images and their augmented versions. The loss is calculated and used to find optimal parameters

ϕ and ω of the trunk and MLP according to (c.f. [8]):

$$\text{InfoNCE}(\phi, \omega) = \arg\min_{\phi,\omega} -\log \frac{e^{sim(z_i,z_j)}}{\sum_{k=1}^{2N} \mathbb{1}_{[k \notin (x_i,x_j)]} e^{sim(z_i,z_k)/\tau}} \tag{3}$$

where

- ϕ and ω are the parameters of the CNN trunk and the MLP, respectively, which need to be learned during training.
- N is the batch size, and $2N$ is the size of the augmented batch.
- z_k represents the feature vectors of the remaining batch images.
- $sim(z_i, z_j)$ is the similarity metric between the feature vectors. It is often computed as the dot product of the feature vectors divided by the product of their norms, which is equivalent to measuring the cosine similarity.
- $\mathbb{1}_{[k \notin (x_i,x_j)]} e^{sim(z_i,z_k)/\tau}$ is an indicator function that equals 1 if k does not belong to the set containing x_i and x_j, and 0 otherwise. This ensures that x_i and x_j are not compared to themselves during the loss calculation.
- τ is a hyperparameter that acts as a temperature parameter, controlling how strongly the feature vectors z_i and z_k are pushed apart in the latent space. It is analogous to the temperature parameter used in a softmax function during training.

By optimizing the SimCLR model with the InfoNCE loss, the CNN trunk and MLP learn to create powerful and transferable image representations, which can be used in downstream tasks like image classification or object detection without the need for labelled data.

DINO is an innovative self-supervised learning method that takes a different approach compared to traditional methods like SimCLR. Instead of using convolutional neural networks (CNNs), DINO employs visual transformers for processing images X and their augmented versions $\hat{X}$. In contrast to SimCLR, which considers multiple images in the batch for comparison, DINO only compares each image with its augmentation within the cost function.

DINO uses two models: a student model $g_{\omega,s}$ and a teacher model $g_{\omega,t}$. These models process the images X and $\hat{X}$ to generate feature representations. Both models are visual transformers, a type of architecture that has shown great success in capturing complex image patterns, and provide good feature capturing capabilities both in spatial and global context.

Before comparing the feature representations, the teacher model $g_{\omega,t}$ processes the original images X to generate the feature vectors $z_t = g_{\omega,t}(X))$. These feature vectors are centred using the batch mean, which helps to remove any biases and normalize the representations. Both the feature vectors of the student model $z_s = g_{\omega,s}(X)$ and the teacher model z_t are sharpened using the Softmax function with an additional temperature parameter τ. The Softmax function amplifies the differences between class probabilities, leading to more discriminative features. The temperature parameter τ controls the concentration of the

probability distribution, where higher values result in softer probabilities and vice versa.

The loss is calculated on the probabilities p_t and p_s as in Caron et al. [7]:

$$H = -p_t \cdot \log p_s \tag{4}$$

The objective of the loss function is to iteratively match the probability distributions of the student and teacher models. By minimizing this cross-entropy loss, the student model learns to mimic the teacher model's predictions and aims to achieve similar probabilities for corresponding images and their augmentations.

Another difference between DINO and contrastive SSL like SimCLR is the use of an asymmetric weight update that prevents mode collapse of the extracted features. During training, the student model is updated using standard backpropagation with stochastic gradient descent (SGD) or other optimization algorithms. The goal is to minimize the cross-entropy loss between the teacher's and student's probabilities. The teacher model, on the other hand, is updated differently. Instead of using large batches or memory banks for negative samples as in contrastive methods, DINO employs an elegant solution by updating the teacher model using the exponential moving average (EMA) of the student model's parameters. This technique helps to stabilize and improve the performance of the teacher model over time. By slowly updating the teacher model to follow the student model's latest state, DINO creates a smoother and more reliable teacher model for guiding the student's learning process.

In summary, DINO's setup and cost function leverage visual transformers, distillation, and iterative matching of probability distributions to learn powerful and transferable image representations in a self-supervised manner. This approach presents a promising alternative to CNN-based methods, demonstrating the effectiveness of visual transformers in addressing the challenges of self-supervised learning with impressive performance results.

2.2 Dataset and Training Details

We use data from the PANDA grand challenge, which contains core needle biopsy (CNB) images taken from Radboud (n = 5160), and Karolinska (n = 5456) hospitals. In total, n = 10616 images are available including their annotated primary and secondary GGs and the GS, as well as the ISUP grade groups (IGGs). Furthermore, masks are provided to split the images into tumour, stromal and non-tumour regions, and in the Radboud cases even GG masks. For the Radboud data, the GS distribution is relatively balanced except for GS 10 (2% share), whereas Karolinska data mostly contains cases of low- to medium severity. After cropping and filtering the CNB images, the l_1 (20× magnification at half resolution) region-annotated (:= ROI) training and test set contains 94678 and 31560, and the non-ROI dataset contains 168256 l_2 (20× magnification at quarter resolution) and 157740 l_1 images.

We test two different SSL methods as described above, with residual Networks (ResNet-18 & 50) [14], as suggested in the PANDA challenge, and a

ViT-B-16 for small datasets as suggested by Lee et al. [16]. Additionally, other frameworks that use a joint embedding architecture, namely SwAV [6], BYOL [12], and MOCO [9], have been tested. As their results were inferior to SimCLR and comparable to DINO, we only present the results of those two methodically diverse approaches. For training, we use PyTorch (1.11.0), Python (3.9) and the training frameworks fastAI (2.7.11) [15] and VISSL (0.1.6) [11]. All parameters are tuned using a grid search. Tab. 1 provides an overview of important augmentation techniques and projection heads, as well as training- and hyperparameters extracted from the grid search. For DINO, using multiple augmentations produced instable training behaviour. Thus, as suggested in [2], we reduced the number of augmentations to reduce the pre-text task complexity. Finally, we only used RandomCrop.

Table 1. Parameter configurations and setup of the SSL algorithms used in this study.

Algorithm	Augmentations	Hyperparameters	Training Parameters	Model
SimCLR	RandomCrop, RandomFlip, ColorDistortion, Gaussian Blur	temperature: 0.7	multi learning rate: • base: 5e-4, • linear: [1e-4, 16e-4], • cosine: [16e-4, 16e-7] SGD/ranger optimizer: • weight decay: 1e-6, • momentum 0.9, • batch size: 32	ResNet-18/50 + MLP: [nc, nc], [nc, 128] nc=512 / 2048
DINO	RandomCrop	teacher τ: [0.01 - 0.02] student τ: 0.4 ema center: 0.9	multi learning rate: • base: 0.05, • linear: [1e-6, 2e-4], • cosine: [2e-4, 1e-6] AdamW optimizer: • weight decay: 1e-6, • batch size: 32	SmallVit-B16 + MLP: [384, 384], [384, 32]

3 Experiments

To evaluate the potential of the SSL-trained feature encoders, we run multiple downstream tasks. The results are presented in this section. The models are trained using an NVIDIA P100 and V100 card. SSL pretraining takes the most memory and processing time of 28 GB and 96 hours, as our setup only allows creating batches up to size 32 for l_1 images, and therefore convergence was only observed after a few hundreds of epochs (c.f. [8]). The supervised models require around quarter the processing time of SSL and converge at around epoch 30. Feature aggregation is used for models in the IGG downstream task and omitted in patch-wise downstream tasks.

Table 2. Overview: Configuration of the self-supervised (SSL) and supervised (SUP) experiments conducted in this study.

No.	Training Type	Network Architecture	Concatenated Features	Evaluation Method
1	SSL	SimCLR (ResNet-18)	No	Embedding Visualization
2	SSL	DINO	No	Embedding Visualization
3	SUP	ResNet-18	No	Embedding Visualization
4	SSL	SimCLR (ResNet-18)	No	Patch Downstream Classification
5	SSL	DINO	No	Patch Downstream Classification
6	SUP	ResNet-18	No	Patch Downstream Classification
7	SSL	SimCLR (ResNet-18)	No	Slide Downstream Classification
8	SSL	SimCLR (ResNet-18)	Yes	Slide Downstream Classification
9	SUP	ResNet-18	Yes	Slide Downstream Classification
10	SUP	ResNet-50	Yes	Slide Downstream Classification

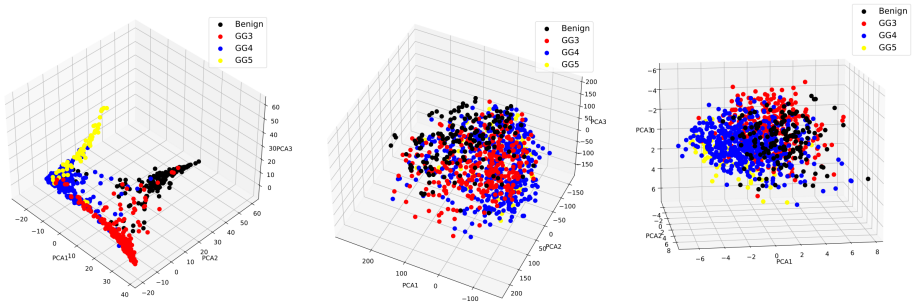
We conducted multiple SSL trainings using different algorithms and downstream tasks. In addition, the models were trained in a supervised fashion, to get a fair comparison. Tab. 2 gives an overview of all conducted experiments. The data for each experiment was structured as in Tab. 3.

3.1 Qualitative Analysis

To get an initial understanding of SSL's capabilities to extract features that align with the GGs, we first perform qualitative analysis using the principal component analysis (PCA) [17] of the feature vectors. For this first downstream task, we only use the Radboud part of the data, as stain differences among hospitals are known to hinder the model from achieving good results. We thus decided to exclude this influence, as stain normalization is not the focus of this study. If the model extracts features that correlate with known prognostic morphologies, the scatter plot of the features' PCA should show separable clusters for each

Table 3. Data used for the individual experiments, including label amount. For scenarios where patches are processed (1-6) the number of labels equals the input images, and for WSI-processing approaches (7-10) 16 input images equal one label.

No.	Image Size	Quality	No. Training Labels	No. Downstream Labels
1-2	128×128	l1	0	-
3	128×128	l1	94678	-
4-5	128×128	l1	0	5917
6	128×128	l1	94678	-
7	128×128	l2	0	657
8	128×128	l2	0	657
9	128×128	l2	10516	-
10	128×128	l1	9858	-



a) Supervised Model **b)** DINO Model **c)** SimCLR Model

Fig. 2. Scatter plots of the first three PCA main components of patch features generated by the supervised, DINO and SimCLR feature encoders.

of the annotated labels. Fig. 2 shows the scatter plot. For the model trained in a supervised fashion, expectations are met. Except for a few cases of low severity which are confused with benign tissue, as well as some confusion between individual GGs, all labels seem to produce separable representations. Yet, the fact that variance among the features seems to be low may be an indicator of missing robustness, as the PCa's heterogenous morphologies should be reflected by the extracted features. The DINO model only produces very rudimentary clusters. In the left part of the plot a cluster of benign tissue can be found which appears to be separable from tumour region patches. Overall, however, a great degree of entangling is observed. This accounts especially for GG5 patches, that can't be distinguished from any of the classes. While the GG5 patches have the lowest class share, this behaviour likely refers to weak features, as at least no overlap should happen with the benign patches (c.f. Fig. 2 a). The SimCLR also shows some entangling but generally appears to extract meaningful features, as clusters of the referred GGs can be observed. In contrast to the DINO model, GG5 shows almost no confusion with the benign tissue and also has clearly separable samples. For the remaining tissue samples, we observe a mix between separable samples and entangled ones. From the qualitative analysis, it was already getting obvious that supervised training provides the best feature extractor, while DINO will likely fail to produce features for detection of the highest severity (GG5), and SimCLR probably achieves plausible but mid-tier results.

3.2 Patch-wise Performance Analysis

To further investigate the effects of SSL training, we analyse the downstream performance using a logistic regression to predict individual patches' GGs. Results are created as the mean of a 3-fold leave-on-out cross validation using the features generated from the ROI test data. Tab. 4 shows the cross-validation performance results as indicated by mean class balanced accuracy and quadratic weighted kappa (BA; QWK), tumour vs. non-tumour balanced accuracy and quadratic

Table 4. Mean patch-wise performance of the different models, in a 3-fold leave-one-out cross validation. The labels refer to benign (B), and Gleason grade 3, 4 and 5 (GG3-5).

Model	resolution	BA	QWK	BBA	BQWK	$f1_B$	$f1_{GG3}$	$f1_{GG4}$	$f1_{GG5}$
Supervised	128^2px@l_1	0.83	0.89	0.94	0.87	0.89	0.89	0.91	0.69
SimCLR	128^2px@l_1	0.63	0.68	0.83	0.65	0.71	0.79	0.78	0.38
DINO	128^2px@l_1	0.52	0.59	0.79	0.59	0.66	0.59	0.73	0.11

weighted kappa (BBA; BQWK), and $f1$ scores of the individual classes. As expected, the DINO model shows the worst results, with a balanced accuracy (BA) of 0.52 and quadratic weighted kappa (QWK) of 0.59. Especially, the $f1$ score of the GG5 patches was lower than random guessing with 0.11. For the supervised model, BA of 0.83 and QWK of 0.89 was achieved. The GG5 show the lowest $f1$ score here as well. SimCLR performs better than the DINO model, while also struggling to classify GG5 patches correctly. Binary BA (BBA) indicates that most false positives arise from confusing tumour classes with each other. In total, the SimCLR performance slightly improves the results achieved in our earlier works (c.f., [5]) using autoencoders, but still is outperformed by the supervised pendant.

3.3 WSI-wise Performance Analysis

While the results achieved in patch-wise classification were promising, they are of less practical relevance, as in a real-world setup predictions need to be done for the WSI rather than individual patches. This task is more challenging, as the network also needs to understand the context of a single patch within its WSI. We thus perform a second classification downstream task to predict the WSI's IGG using the non-ROI l_2 dataset and the same preprocessing, validation strategy and split ratio as before. Multiple setups are evaluated, as presented in Tab. 5. For the SSL models, the encoder part was frozen after training and only the aggregation and projection head are fine-tuned. For fine-tuning and training of the supervised pendants, kappa loss [25] yields the best results. For the SSL part, we focus on the SimCLR model, as DINO didn't appear to deliver useful results.

In general, training by SimCLR without feature aggregation achieves inferior results. For the QWK, the best results at lowest magnification are achieved by the supervised model. The results are in accordance with the PANDA challenge's baseline solution. Tab. 5 suggests that significant differences between the $f1$ scores of the individual IGG can be found. The worst results are achieved for the IGGs that contain the most heterogenous patterns. This seems plausible, as the training method enforces the network to not only extract meaningful features, but also to decide which features in combination are connected to a certain IGG. Thus, better $f1$ scores are achieved, if less feature variance is present. As the QWK not necessarily captures this behaviour, we also compare for balanced

Table 5. Mean WSI-wise performance of the different models, in a 3-fold leave-one-out cross validation. The labels refer to benign (B) and the IGGs 1-5 (I1-I5).

Model	resolution	BA	QWK	BBA	BQWK	$f1_B$	$f1_{I1}$	$f1_{I2}$	$f1_{I3}$	$f1_{I4}$	$f1_{I5}$
SimCLR ResNet-18	128^2px@l_2	0.41	0.57	0.75	0.47	0.63	0.44	0.29	0.24	0.28	0.51
SimCLR ResNet-18 + concat	128^2px@l_2	**0.51**	0.65	**0.85**	**0.64**	**0.75**	0.54	**0.42**	0.33	**0.46**	0.57
Supervised ResNet-18 + concat	128^2px@l_2	0.50	**0.76**	0.83	0.62	0.74	0.54	0.30	**0.34**	0.25	**0.62**
Supervised ResNet-50 + concat	128^2px@l_2	0.50	0.74	0.83	**0.64**	**0.75**	**0.56**	0.34	0.28	0.40	0.58

accuracy and individual $f1$ scores. Here, the SimCLR model achieves the best results most of the time, whereas some results are shared with the supervised ResNet-50. The most prominent advantage can be seen for IGGs two and four.

As l_1 images are known to create better accuracies in supervised training (c.f. [3]), we repeat the experiment with non-ROI l_1 images. In this case, similar to the PANDA challenge's results, BA of 0.7+ and QWK of 0.85+ were achieved when using a supervised ResNet-50. For SimCLR, We found that batch size seems to be a critical factor here, but couldn't fully investigate the limits of SSL performance, as the required amount of graphical memory is exceeding our technical capabilities. The results, even though of low BA and QWK, still indicated a balanced behaviour as for the l_2 images.

4 Discussion

In this paper, we showed, that SSL pretraining may be a promising method, to create state-of-the-art PCa grading models, with significantly less effort in annotating cases beforehand. Our models achieve better results in qualitative and quantitative analysis than earlier autoencoder-based works.

Current supervised models are limited by the accuracy of the pathologists themselves (e.g., for PANDA the accuracy of the annotating pathologists against expert consensus, similar to the CNNs, was only at 72%). Our results indicate, that SSL pretraining has the potential to extract prognostic features, which are unaffected by this issue. SSL thus provides a platform to combine morphological information with follow-up-based endpoints as BCR or DoD to directly identify prognostic features after SSL pretraining.

When training the models, we observed, that various hyperparameters such as the batch size and image resolution can have a significant impact on the results. Thus, to fully evaluate the limits of the presented approach, future work should

apply a more sophisticated hyperparameter tuning and architecture search. Furthermore, one main issue of the IGG prediction seems to be directly connected to the concatenation-based training. For the SSL approach, this also comes with the downside of high graphical memory consumption. We thus propose, to investigate better strategies for combining various features in future works.

Another important research direction could also lie in using SSL as a data exploration tool. Even though the SSL pretraining reduced the amount of required labels to achieve state-of-the-art classification results to 25% compared to the supervised approach, a total amount of 657 annotated patient cases is still a lot. Our qualitative analyses showed that morphological groups are identified by the algorithm. Hence, SSL may have the capability to reduce the amount of annotated cases even further, by labelling, e.g., the most uncertain cases in an expert-in-the-loop approach.

We conclude that, given the promising results of our work, SSL deserves to get more attention in histopathology. This especially accounts for creating an expert-in-the-loop system, which could also help pathologists to gather new knowledge from the AI, rather than only providing it.

References

1. Arvaniti, E., Fricker, K.S., Moret, M., Rupp, N., Hermanns, T., Fankhauser, C., Wey, N., Wild, P.J., Rüschoff, J.H., Claassen, M.: Automated gleason grading of prostate cancer tissue microarrays via deep learning. Scientific Reports **8**(1) (Aug 2018)
2. Balestriero, R., Ibrahim, M., Sobal, V., Morcos, A., Shekhar, S., Goldstein, T., Bordes, F., Bardes, A., Mialon, G., Tian, Y., Schwarzschild, A., Wilson, A.G., Geiping, J., Garrido, Q., Fernandez, P., Bar, A., Pirsiavash, H., LeCun, Y., Goldblum, M.: A cookbook of self-supervised learning. CoRR **abs/2304.12210** (2023)
3. Bulten, W., Kartasalo, K., Chen, P.H.C., Ström, P., Pinckaers, H., Nagpal, K., Cai, Y., Steiner, D.F., van Boven, H., Vink, R., van de Kaa, C.H., van der Laak, J., Amin, M.B., Evans, A.J., van der Kwast, T., Allan, R., Humphrey, P.A., Grönberg, H., Samaratunga, H., Delahunt, B., Tsuzuki, T., Häkkinen, T., Egevad, L., Demkin, M., Dane, S., Tan, F., Valkonen, M., Corrado, G.S., Peng, L., Mermel, C.H., Ruusuvuori, P., Litjens, G., Eklund, M., Brilhante, A., Çakır, A., Farré, X., Geronatsiou, K., Molinié, V., Pereira, G., Roy, P., Saile, G., Salles, P.G.O., Schaafsma, E., Tschui, J., Billoch-Lima, J., Pereira, E.M., Zhou, M., He, S., Song, S., Sun, Q., Yoshihara, H., Yamaguchi, T., Ono, K., Shen, T., Ji, J., Roussel, A., Zhou, K., Chai, T., Weng, N., Grechka, D., Shugaev, M.V., Kiminya, R., Kovalev, V., Voynov, D., Malyshev, V., Lapo, E., Campos, M., Ota, N., Yamaoka, S., Fujimoto, Y., Yoshioka, K., Juvonen, J., Tukiainen, M., Karlsson, A., Guo, R., Hsieh, C.L., Zubarev, I., Bukhar, H.S.T., Li, W., Li, J., Speier, W., Arnold, C., Kim, K., Bae, B., Kim, Y.W., Lee, H.S., and, J.P.: Artificial intelligence for diagnosis and gleason grading of prostate cancer: the PANDA challenge. Nature Medicine **28**(1), 154–163 (Jan 2022)
4. Bulten, W., Litjens, G.: Unsupervised prostate cancer detection on h&e using convolutional adversarial autoencoders. CoRR **abs/1804.07098** (2018)
5. Bulten, W., Pinckaers, H., van Boven, H., Vink, R., de Bel, T., van Ginneken, B., van der Laak, J., van de Kaa, C.H., Litjens, G.: Automated deep-learning system

for gleason grading of prostate cancer using biopsies: a diagnostic study. The Lancet Oncology **21**(2), 233–241 (Feb 2020)
6. Caron, M., Misra, I., Mairal, J., Goyal, P., Bojanowski, P., Joulin, A.: Unsupervised learning of visual features by contrasting cluster assignments. In: Proceedings of the 34th International Conference on Neural Information Processing Systems. NIPS'20, Curran Associates Inc., Red Hook, NY, USA (2020)
7. Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., Joulin, A.: Emerging properties in self-supervised vision transformers. CoRR **abs/2104.14294** (2021)
8. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A simple framework for contrastive learning of visual representations. CoRR **abs/2002.05709** (2020)
9. Chen, X., Fan, H., Girshick, R., He, K.: Improved baselines with momentum contrastive learning. CoRR **abs/2003.04297** (2020)
10. Gidaris, S., Singh, P., Komodakis, N.: Unsupervised representation learning by predicting image rotations. CoRR **abs/1803.07728** (2018)
11. Goyal, P., Duval, Q., Reizenstein, J., Leavitt, M., Xu, M., Lefaudeux, B., Singh, M., Reis, V., Caron, M., Bojanowski, P., Joulin, A., Misra, I.: VISSL. https://github.com/facebookresearch/vissl (2021)
12. Grill, J.B., Strub, F., Altché, F., Tallec, C., Richemond, P.H., Buchatskaya, E., Doersch, C., Pires, B.A., Guo, Z.D., Azar, M.G., Piot, B., Kavukcuoglu, K., Munos, R., Valko, M.: Bootstrap your own latent a new approach to self-supervised learning. In: Proceedings of the 34th International Conference on Neural Information Processing Systems. NIPS'20, Curran Associates Inc., Red Hook, NY, USA (2020)
13. He, K., Fan, H., Wu, Y., Xie, S., Girshick, R.: Momentum contrast for unsupervised visual representation learning. CoRR **abs/1911.05722** (2019)
14. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. CoRR **abs/1512.03385** (2015)
15. Howard, J., Gugger, S.: fastai: A layered API for deep learning. CoRR **abs/2002.04688** (2020)
16. Lee, S.H., Lee, S., Song, B.C.: Vision transformer for small-size datasets. CoRR **abs/2112.13492** (2021)
17. Maćkiewicz, A., Ratajczak, W.: Principal components analysis (PCA). Computers & Geosciences **19**(3), 303–342 (Mar 1993). https://doi.org/10.1016/0098-3004(93)90090-r, https://doi.org/10.1016/0098-3004(93)90090-r
18. Misra, I., van der Maaten, L.: Self-supervised learning of pretext-invariant representations. CoRR **abs/1912.01991** (2019)
19. Nagpal, K., Foote, D., Tan, F., Liu, Y., Chen, P.H.C., Steiner, D.F., Manoj, N., Olson, N., Smith, J.L., Mohtashamian, A., Peterson, B., Amin, M.B., Evans, A.J., Sweet, J.W., Cheung, C., van der Kwast, T., Sangoi, A.R., Zhou, M., Allan, R., Humphrey, P.A., Hipp, J.D., Gadepalli, K., Corrado, G.S., Peng, L.H., Stumpe, M.C., Mermel, C.H.: Development and validation of a deep learning algorithm for gleason grading of prostate cancer from biopsy specimens. JAMA Oncology **6**(9), 1372 (Sep 2020)
20. Noroozi, M., Favaro, P.: Unsupervised learning of visual representations by solving jigsaw puzzles. CoRR **abs/1603.09246** (2016)
21. Oord, A.v.d., Li, Y., Vinyals, O.: Representation learning with contrastive predictive coding. CoRR **abs/1807.03748** (2018)
22. Singhal, N., Soni, S., Bonthu, S., Chattopadhyay, N., Samanta, P., Joshi, U., Jojera, A., Chharchhodawala, T., Agarwal, A., Desai, M., Ganpule, A.: A deep learning system for prostate cancer diagnosis and grading in whole slide images of core needle biopsies. Scientific Reports **12**(1) (Mar 2022)

23. Ström, P., Kartasalo, K., Olsson, H., Solorzano, L., Delahunt, B., Berney, D.M., Bostwick, D.G., Evans, A.J., Grignon, D.J., Humphrey, P.A., Iczkowski, K.A., Kench, J.G., Kristiansen, G., van der Kwast, T.H., Leite, K.R.M., McKenney, J.K., Oxley, J., Pan, C.C., Samaratunga, H., Srigley, J.R., Takahashi, H., Tsuzuki, T., Varma, M., Zhou, M., Lindberg, J., Lindskog, C., Ruusuvuori, P., Wählby, C., Grönberg, H., Rantalainen, M., Egevad, L., Eklund, M.: Artificial intelligence for diagnosis and grading of prostate cancer in biopsies: a population-based, diagnostic study. The Lancet Oncology **21**(2), 222–232 (Feb 2020)
24. Tolkach, Y., Dohmgörgen, T., Toma, M., Kristiansen, G.: High-accuracy prostate cancer pathology using deep learning. Nature Machine Intelligence **2**(7), 411–418 (Jul 2020)
25. Vaughn, D., Justice, D.: On the direct maximization of quadratic weighted kappa. CoRR **abs/1509.07107** (2015)
26. Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., Manzagol, P.A.: Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. J. Mach. Learn. Res. **11**, 3371–3408 (dec 2010)
27. Wu, Z., Xiong, Y., Yu, S., Lin, D.: Unsupervised feature learning via non-parametric instance-level discrimination. CoRR **abs/1805.01978** (2018)
28. Yan, J., Chen, H., Li, X., Yao, J.: Deep contrastive learning based tissue clustering for annotation-free histopathology image analysis. Computerized Medical Imaging & Graphics **97**, N.PAG–N.PAG (2022)
29. Zbontar, J., Jing, L., Misra, I., LeCun, Y., Deny, S.: Barlow twins: Self-supervised learning via redundancy reduction. CoRR **abs/2103.03230** (2021)