Multi-Prototype Few-shot Learning in Histopathology

Jessica Deuschel[1] Daniel Firmbach[1] Carol I. Geppert[2] Markus Eckstein[2]
Arndt Hartmann[2] Volker Bruns[1] Petr Kuritcyn[1] Jakob Dexl[1] David Hartmann[1]
Dominik Perrin[1] Thomas Wittenberg[1] Michaela Benz[1]

[1]Fraunhofer Institute for Integrated Circuits IIS
Am Wolfsmantel 33, 91058 Erlangen, Germany
jessica.deuschel@iis.fraunhofer.de

[2] Institute of Pathology, University Hospital Erlangen, Friedrich-Alexander-University Erlangen-Nuremberg
Krankenhausstr. 8-10, 91054 Erlangen, Germany
carol.geppert@uk-erlangen.de

Abstract

The ability to adapt quickly to a new task or data distribution based on only a few examples is a challenge in AI and highly relevant for various domains. In digital pathology, slight variations in the scanning and staining process can lead to a distribution shift that provokes significant performance degradation of classical neural networks for tasks like tissue cartography where a reliable classification is essential. To overcome this problem, we propose a few-shot learning technique, specifically a k-means extension of Prototypical Networks, to train a highly flexible model that adapts to new, unseen scanner data based on only a few examples. We evaluate our approach on a multi-scanner database comprising a total amount of 356 annotated whole slide images digitized by a base scanner for training and additional five different scanners for evaluation. We verify our method's effectiveness by comparing it to a classically trained benchmark and Prototypical Networks, both trained on the same data. A particular focus for us is to investigate the support set, used for adapting the prototypes, to provide recommended actions for digital pathology. The best results are obtained by employing multiple prototypes per class, calculated from a distributed support set, and domain-specific data augmentation. This results in 86.9 - 88.2% accuracy for a classification task of seven tissue classes on unseen, shifted data from the automated scanners, which is almost equal to the accuracy on the in-distribution data of 89.2%.

1. Introduction

Ongoing progress in artificial intelligence in computer vision, along with the introduction of digital pathology, has opened up numerous possibilities for supporting pathological examinations. Microscopic tissue sections can be digitized as whole slide images (WSIs), preprocessed, and analyzed. The digitized form makes it possible to apply deep learning techniques, for example, categorizing different tissue types and cells or predicting medical endpoints [20]. However, the deployment of these methods is challenging: The digitization process is highly dependent on the microscopic scanner and heterogeneous preprocessing steps (*e.g.* staining protocols) that both differ from clinic to clinic. Differences between scanners from diverse manufactures impact the image color (such as hue, brightness, and contrast) and resolution. This variance leads to a shift in the data distribution and degrades the performance of common deep learning classifiers, as they depend strongly on the distribution of the training data [13, 12].

Therefore, our goal is to train a highly flexible model that can adapt to this distribution shift imposed by new scanner data. Domain adaptation and specifically few-shot learning addresses this challenge by permitting fast adaptation to a new domain, even with only a limited amount of labeled data available [19, 5]. Research in this area proposes a wide range of approaches for the task of image classification. They range from data-driven methods (*e.g.* augmentation techniques) to algorithm-driven methods (*e.g.* [5, 2, 3, 21, 26]) that mainly focus on fine-tuning, to model-driven methods that classify new data by comparison in an

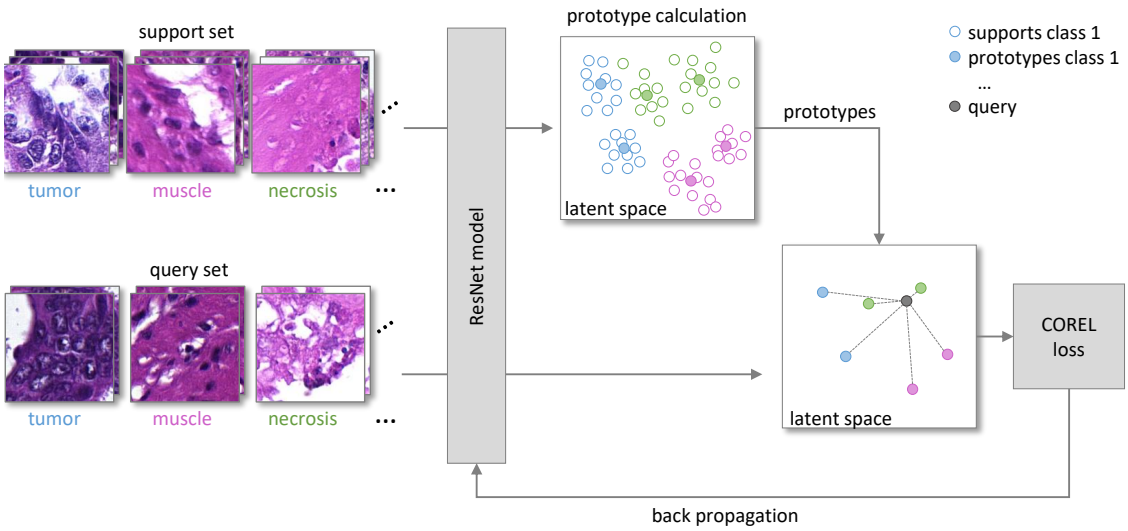

Figure 1. Training procedure of Multi-ProtoNets. The input for one episode consists of a support set of 20 image patches per class and a query set of 5 image patches per class. We train with seven classes, three are depicted. The supports and queries are passed through a ResNet model that transfers all inputs into a 512-dimensional latent space. We use the support set to calculate multiple prototypes per class based on a k-means clustering algorithm (here depicted with $k = 2$ prototypes), where the prototypes are the cluster centers. Next, we determine the distance of each query to the prototypes (depicted for only one query), calculate the COREL loss and conduct a back propagation using Adam optimizer.

embedding (or latent) space [25].

Data augmentation is a popular technique, not solely in the domain of few-shot learning, but applied in many applications and domains [27, 10]. It enables the inclusion of a priori knowledge: an exposure of the hue value can for example resemble data from a different scanner. In digital pathology, data augmentation techniques such as H&E color stain augmentation and spatial filters (*e.g.* rotation and scaling) have been used to mimic variation in the data and improve the performance of learning algorithms [13, 23].

In our research we focus on model-driven few-shot methods, specifically embedding learning, which, unlike algorithm-driven methods, do not require retraining. Instead, these methods exploit a lower-dimensional embedding of the data, where samples are clustered based on their similarity [25]. New samples can also be compared in this embedding space. Representatives of this method are Siamese Networks [11], Relation Networks [22] and Matching Networks [24]. One of the most popular methods, due to its simplicity, is Prototypical Networks [19], proposed by Snell *et al*., which calculates the center (the prototype) of each class in the embedding space and assigns the class of the nearest prototype to each new example.

Snell *et al*. use only one prototype per class and argue for its sufficiency, opposed by Mensink *et al*. who suggest multiple class centroids by applying a k-means algorithm directly on the input space [15]. We think that it is worth following up on this idea, however with the application of k-means on the embedding space of a deep neural network instead of the input space.

Therefore we propose Multi-ProtoNets, a k-means extension of Prototypical Networks that uses multiple prototypes to represent each class and thus learns a more differentiated embedding space that allows for disconnected class representations. Our approach is similar to [18], where a density representation in the embedding space is learned by applying k-means clustering. However, they do not consider this approach in the few-shot setting and therefore do not apply an episodic training procedure. Others have used a soft k-means [17] and Sinkhorn k-means [8] in few-shot learning; the former to include unlabelled data in the training but not for multiple class representatives. The latter only applies k-means for calculating multiple prototypes during test time. We, however, show that it is beneficial to already train with k-means to be able to learn a more sophisticated, possibly non-connected cluster representation. Similar to our approach, Infinite Mixture Prototypes [1] also generate multiple clusters per class, even without having to specify a fixed number of class clusters, but with the challenge of estimating the distance threshold parameter. This method has already been applied for Relation Networks [14] and showed the positive effect of learning multiple prototypes. Our method is much simpler to implement and more intuitive because we apply the distance metric directly on the embedding space without the need to train an additional relational layer. Also related to our approach multiple pro-

totypes have been used in the medical domain, for skin disease identification [16], where k-means is only used at the start of an epoch as an initialization of the prototypes, whereas we use it in every episode. Our work also differs from the others as we work with the COREL loss which facilitates clustering. To the best of our knowledge, no other method in few-shot learning exists at the moment that extends Prototypical Networks by learning with multiple prototypes based on a k-means approach and COREL loss on the embedding space of a neural network.

We extensively test our method's ability to adapt to five shifted data sets, digitized by different scanners that vary in the scanning and stitching process. We experimentally demonstrate the positive effect of a multiple cluster representation in the embedding space, as given by Multi-ProtoNets, compared to Prototypical Networks and a classic baseline. This effect is also higher if multiple prototypes are already used during training. We further investigate the influence of different numbers of clusters ($k = \{1, \dots, 5\}$) per class, where we achieve the best results with $k = 3$ clusters. Our few-shot approach allows the adaptation of a model to new data on the basis of a few examples, which are used to determine new prototypes. However, how much does the quality of the adapted model depend on the variance of the data used to recalculate the prototypes? To answer this question, we systematically vary the number of WSIs and their combinations from which a set of image patches is chosen to adapt the model and investigate how many WSIs are needed. Finally, we continue to improve the performance by applying domain-specific data augmentation.

Overall, this work contributes to providing recommendations for the design of a robust few-shot approach for digital pathology, allowing easy adaptability to domain shifts introduced *e.g.* by different scanners.

2. Prototypical Networks for Few-Shot Learning with Multiple Prototypes

We consider a training data set $\mathcal{D} = (\mathbf{x}_i, y_i)_{i=1}^N$ of N train-label pairs with input data $\mathbf{x}_i \in \mathbb{R}^n$ and class label $y_i \in \{1, \dots, C\}$. We use a model $f_\theta : \mathbb{R}^n \to \mathbb{R}^d$, in our case a ResNet34 architecture [7], with parameters θ to obtain a representation $\mathcal{R} = (f_\theta(\mathbf{x}_i), y_i)_{i=1}^N$ of the data in the latent space.

In few-shot learning an episodic training procedure is common, referred as N^{c}-way N^{s}-shot: For each episode, a subset $\mathcal{C} \subseteq \{1, \dots, C\}$ of N^{c} classes is chosen at random. For each class $c \in \mathcal{C}$ we choose a support set $\mathcal{S}_c \subseteq \mathcal{R}$ of size N^{s} and a query set $\mathcal{Q}_c \subseteq \mathcal{R} \setminus \mathcal{S}_c$ of size N^{q} where all examples of both $\mathcal{S}_c$ and $\mathcal{Q}_c$ correspond to class label c. The support set is used to calculate class representatives or prototypes; the prediction and loss calculation is done on the query set based on the proximity to the closest prototype in the latent space. For a visualization of the training process we refer to Figure 1.

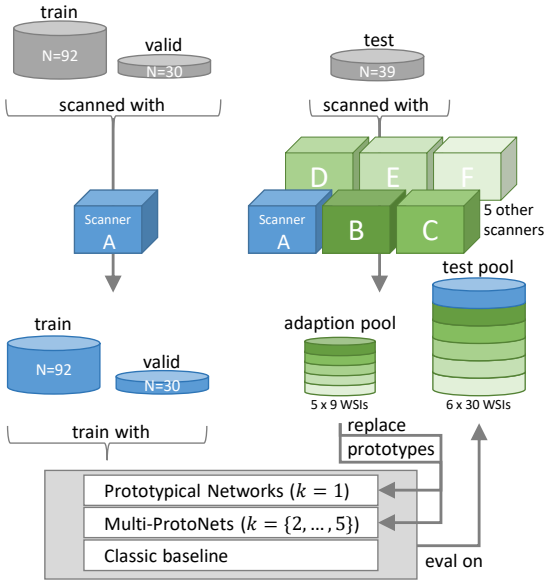

Figure 2. Overview of our training and evaluation procedure. Based on the training data digitized by scanner A (MIDI) three different methods are trained: Prototypical Networks, Multi-ProtoNets with $k \geq 2$, and a classic baseline. The test set is digitized by six different scanners and the resulting WSIs are divided into a set of nine slides, reserved for prototype adaptation for both few-shot learning methods, and a set of 30 slides to test their ability to adapt to the introduced data shifts.

2.1. Prototype Calculation

In contrast to Prototypical Networks which rely on the assumption that the data can be translated into an embedding space where each class is represented by a sufficiently convex and connected cluster to be captured by a single prototype, we follow a more general approach: We calculate multiple prototypes per class based on a k-means clustering on the embedding of the corresponding support set and therefore allow for a more distributed class representation. Clustering, specifically k-means, is a powerful tool to group similar examples in a fully unsupervised manner [6], which we apply within the few-shot training setting. Our goal is to divide the embedding space into k clusters for each class c, so that the distance of each example to the closest cluster center $\boldsymbol{\mu}_j$, the representative of cluster $j \in \{1, \dots, k\}$, becomes minimal. k-means clustering repeatedly assigns the supports in $\mathcal{S}_c$ to k clusters, based on the lowest squared $\mathcal{L}_2$ distance, and updates the cluster centers, by taking the mean of the assigned values, until the prototypes no longer change. k-means is proven to converge to at least the local optimum and usually does so within a small number of steps [6].

2.2. Loss Function

As a next step the loss is calculated based on the proximity of the queries $(\mathbf{x}_i, y_i) \in \mathcal{Q}_c$ to their nearest cluster centers $\boldsymbol{\mu}_j^{(y_i)}$ of the corresponding class y_i and to the ones of all classes $\boldsymbol{\mu}_j^{(c)}$, $c \in \mathcal{C}$. Both is considered in the COREL loss [9],

$$\mathcal{L} = \sum_{i=1}^{N^{\mathrm{q}}} \lambda \mathcal{L}_i^{\mathrm{att}} + (1-\lambda)\mathcal{L}_i^{\mathrm{rep}}, \tag{1}$$

which we adapt to the case of multiple prototypes. This loss is a weighted composition of an attractive and a repulsive part,

$$\mathcal{L}_i^{\mathrm{att}} = \gamma \min_{j \in \{1,\dots,k\}} (||f_\theta(\mathbf{x}_i) - \boldsymbol{\mu}_j^{(y_i)}||_2^2), \tag{2}$$

$$\mathcal{L}_i^{\mathrm{rep}} = \log \sum_{c \in \mathcal{C}} e^{-\gamma \min_{j \in \{1,\dots,k\}} (||f_\theta(\mathbf{x}_i) - \boldsymbol{\mu}_j^{(c)}||_2^2)}, \tag{3}$$

weighted with hyper-parameters $\lambda \in (0, 1]$ and $\gamma > 0$. $\mathcal{L}^{\mathrm{att}}$ favors a tighter clustering between members of the same class in the latent space, and $\mathcal{L}^{\mathrm{rep}}$ causes a drift between the classes. It naturally leads to a separable representation in the embedding space [9] and is therefore highly suitable for our combination with *k*-means clustering.

The pseudo-code for our Multi-ProtoNets approach is presented in Algorithm 1.

Algorithm 1 Multi-ProtoNets

Input: data: $\mathcal{D} = (\mathbf{x}_i, y_i)_{i=1}^N$, network with initial parameters θ: $f_\theta : \mathbb{R}^n \to \mathbb{R}^d$

for each episode **do**
  Select classes $\mathcal{C} \subseteq \{1, \dots, C\}$
  for $c \in \mathcal{C}$ **do**
    Select support and query sets $\tilde{\mathcal{S}}_c \subseteq \mathcal{D}$ and $\tilde{\mathcal{Q}}_c \subseteq \mathcal{D} \setminus \tilde{\mathcal{S}}_c$
    Embed supports and queries as $\mathcal{S}_c = \{f_\theta(\mathbf{x}_i), y_i\}_{i=1}^{N^{\mathrm{s}}}$ and $\mathcal{Q}_c = \{f_\theta(\mathbf{x}_i), y_i\}_{i=1}^{N^{\mathrm{q}}}$
    Select k prototypes $\{\boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_k\}$, $\boldsymbol{\mu}_j \in \mathbb{R}^d$ at random out of $\mathcal{S}_c$
    Assign a cluster-label to each instance $\mathbf{x}_i \in \tilde{\mathcal{S}}_c$ based on the proximity to the closest prototype:
    $$r_{ij} = \begin{cases} 1, & \text{if } j = \arg\min_{\kappa \in \{1,\dots,k\}} ||(f_\theta(\mathbf{x}_i) - \boldsymbol{\mu}_\kappa)||_2^2 \\ 0, & \text{otherwise} \end{cases},$$
    $\forall i \in \{1, \dots, N^{\mathrm{s}}\}, \forall j \in \{1, \dots, k\}$
    Create new prototypes:
    $\boldsymbol{\mu}_j \leftarrow \frac{1}{\sum_{i=1}^{N^{\mathrm{s}}} r_{ij}} \sum_{i=1}^{N^{\mathrm{s}}} r_{ij} f_\theta(\mathbf{x}_i)$
  end for
  Calculate the COREL loss based on the queries according to equation 1:
  $$\mathcal{L} = \sum_{i=1}^{N^{\mathrm{q}}} \Big(\gamma \min_j (||f_\theta(\mathbf{x}_i) - \boldsymbol{\mu}_j^{(y_i)}||_2^2) \cdot \lambda + \log \sum_{c \in \mathcal{C}} e^{-\gamma \min_j (||f_\theta(\mathbf{x}_i) - \boldsymbol{\mu}_j^{(c)}||_2^2)} \cdot (1-\lambda)\Big),$$
  where $\mathbf{x}_i \in \tilde{\mathcal{Q}}_c$ and $\boldsymbol{\mu}_j^{(c)}, j \in \{1, \dots, k\}$ prototypes corresponding to class c
  Update θ using Adam optimizer
end for

3. Multiple-Scanner Data

The whole data set (training, validation and test) consists of 356 annotated WSIs from which more than 10 million labeled image patches with a dimension of 224×224 pixels are derived. The WSIs are obtained from a collection of 161 H&E stained colon tissue sections from adenocarcinoma resections. All sections are digitized with the 3DHistech MIDI scanner with a resolution of 0.22 µm per pixel at the University Hospital Erlangen. Within the resulting WSIs of the MIDI scanner, areas have been manually annotated distinguishing between seven tissue types: tumor, muscle tissue, connective combined with adipose tissue, mucosa, mucus, inflammation, and necrosis. Finally, image patches of pixel size 224×224 are extracted from the WSIs and assigned to a tissue class if a patch intersects with a manual annotation by at least 85%. The training database comprises 2,173,515 image patches from 92 tissue sections and the validation database 719,010 patches from 30 tissue sections, all digitized with the 3DHistech MIDI scanner. Patches from a single tissue section are never split across multiple data sets. The number of image patches per class and WSI is limited to 10,000 for these two databases. Based on the remaining 39 tissue sections whose WSIs are not used for training and validation a multi-scanner test database is established. Therefore these tissue sections are additionally digitized with four automated scanners (SCube, Precipoint M8, Hamamatsu Nanozoomer S210, Hamamatsu Nanozoomer S360) and one manual microscope using a real-time stitching software (iSTIX) resulting in a test database of 234 WSIs: six scanner-specific sets each comprising 39 WSIs. Annotations are transferred automatically by rigid registration of the corresponding WSIs. A set of nine WSIs per scanner (corresponding to identical tissue sections) is set aside for the prototype adaptation. The remaining 30 WSIs of each scanner represent the multi-scanner test database. Figure 2 gives an overview of the data sets and what they are used for. The number of patches within the scanner-specific test databases varies between 514,397 and 2,123,364 due to differences in scanner resolution (between 0.17 µm/pixel and 0.35 µm/pixel) and in background detection, as image patches from the background are discarded using a simple threshold approach.

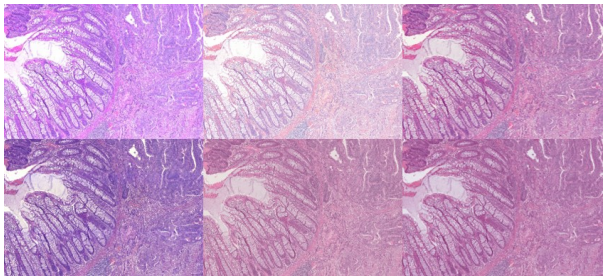

Figure 3. Cutout of a digitized slide scanned with - from left to right - top row: MIDI, iSTIX, M8; bottom row: SCube, S210, S360.

Moreover, there are significant color variations between WSIs obtained with the different scanners (see Figure 3).

4. Experiments

We train various models: a classic baseline as benchmark that does not employ few-shot learning, and five few-shot models based on k-means, for $k = 1, \ldots, 5$ respectively. Note that $k = 1$ is exactly equal to Prototypical Networks and therefore also works as state of the art comparison baseline to our approach. For embedding we choose ResNet34, a Residual Network architecture with 32 convolutional layers arranged in 16 blocks with skip connections, average pooling and a fully connected layer. For few-shot learning we omit the latter as we work directly on the latent space. Prior to our experiments, we investigated different latent space dimensions and obtained the best results with a dimension of 512, which we use in all subsequent experiments. We use Adam optimizer with a learning rate of 0.001 for the classic approach and 10^{-5} for few-shot learning with early stopping. The parameters of the COREL loss are selected by default: $\lambda = \gamma = 0.5$. Similar to [4], we choose a 7-way 20-shot scenario with $N^{\mathrm{q}} = 5$ for training, i.e. all classes are used in each episode. The implementation is written in Python with Tensorflow 2.2.0. All models are trained using the same training and validation database containing only WSIs of the 3DHistech MIDI scanner (see Section 3). For the classic approach we average the results over two training and test runs. The different few-shot models, on the other hand, are averaged over the extensive test runs.

In addition, we repeat our training runs, both the few-shot and the classic approach, with domain-specific data augmentation. Our augmentation consists of variations in hue and saturation as well as additional H&E augmentation [23]. These three augmentations proved to be very effective on histological data, whereas the influence of deviations in resolution seemed to be less critical than in color [13].

k^{train}	1	1	1	3	5
k^{eval}	1	3	5	3	5
Accuracy	0.602	0.630	0.633	0.705	0.705

Table 1. Comparison of the accuracy between models that use multiple prototypes both during training and evaluation, and models that are trained with a single prototype and use multiple prototypes only during evaluation. Trained without augmentation and evaluated on the M8 data set with adapted prototypes from a set of nine slides.

4.1. Multiple Prototypes and Variations in Support Set

The advantage of our few-shot approach is that information from other scanners can be easily utilized without retraining the network: in our experiments nine annotated slides from each scanner ("adaption pool", see Figure 2) are provided to calculate new prototypes adapted to the specific scanner at inference time. We compare two cases: (a) prototypes are derived from the training set (non-adapted case), and (b) prototypes are derived individually for each scanner from the hold-out data set of nine slides (adapted case). In both cases, 1000 supports per class are used except when there are not enough supports for this class available in the data set.

We calculate the prototypes by a k-means clustering and compare different numbers of prototypes ($k = 1, \ldots, 5$) per class. Moreover, we investigate the influence of the support set on the classification performance to answer the following questions: Does it make a difference if we select the same number of supports from different WSIs? Is it better to select the supports from a larger number of WSIs and thus provide more variability in the support set? Hence, we choose the supports from different numbers of slides from the hold-out data set: 3,5,7 and 9. Since our hold-out set contains nine WSIs for each scanner, we evaluate each possible combination, *e.g.* choosing three slides out of nine and selecting the supports randomly from this subset. Only for the subset size of nine there are no possible permutations. In this case we repeat the prototype calculation 40 times with randomly chosen supports from the nine slides. As each experiment in the testing process is conducted at least 40 times, we also report the standard deviation (see Figure 4).

Firstly, we observe that a scanner-specific adaptation of the prototypes has a positive effect compared to a prototype calculation on the original scanner data. This is depicted in Figure 4 for M8 and Scube. The effect is especially striking for M8, where the accuracy increases by at least 19.2 percentage points. On SCube we observe only a slight increase of between 0.4 and 8.1 percentage points. We hypothesize that this is due to the higher similarity between SCube and the original scanner (MIDI) in terms of color representation.

Comparing different choices of k, we find the methods

using multiple prototypes ($k > 1$) to be superior (by up to 10.4 percentage points) compared to using the original Prototypical Networks approach, thus a single prototype per class ($k = 1$). Especially $k = 3$ prototypes yield particularly good results over all scanners. We hypothesize that multiple prototypes provide a better class representation in the embedding space, allowing for a more accurate clustering especially for tissue classes with a strong variance in appearance like *e.g.* tumor. We also note a positive effect on the accuracy when training is done with only one prototype but testing with multiple, as shown in Table 1. As the table indicates, the accuracy increases monotonically with the number of clusters used for training and for testing for the M8 scanner. At some point the accuracy seems to stagnate because there is no difference anymore between five prototypes used for both training and testing and three. Overall, the best accuracy is achieved when multiple prototypes, either three or five, are used for both training and testing.

We further observe that, independent of the number of clusters, the accuracy increases and standard deviation decreases as more WSIs are used for prototype calculation. For $k = 3$ the accuracy gradually increases from 69.3 percentage points for three slides to 70.5 for nine slides and standard deviation decreases by 0.015 for M8. Again the effect is smaller for SCube, where accuracy increases by 0.75 percentage points and standard deviation decreases by 0.007 between three and nine slides for $k = 3$. However, the trend is clearly visible. This is consistent with our assumptions and suggests that, since more slides cause a higher variance in the support set, especially distributed slides make the prototypes more robust.

4.2. Combination of Few-Shot and Data Augmentation

As both few-shot and data augmentation independently enhance the overall performance [13], we expect a further improvement from their combination. Therefore, as depicted in Figure 5, we compare the models that were trained with and without data augmentation. We present the results for a ($k = 3$)-training setting and a testing scenario where nine slides from new scanners are available for prototype adaptation. Notably, augmentation improves the performance over all scanners, both for the classic approach and few-shot learning. For Multi-ProtoNets we register an absolute increase in accuracy of up to 25.1 percentage points and up to 0.274 for the F1-score (see Table 2). Only for MIDI and SCube the effect is moderate, since the performance has already been strong without augmentation. Overall, our results show that data augmentation generates a more robust embedding space, which is also beneficial for few-shot learning. As a final result, for all automated scanners we achieve an accuracy of between 86.9% and 88.2%, similar to that obtained on the original scanner (MIDI). Only on the iSTIX scans the performance drops to 71.8%. This might be due to the lower quality of the WSIs obtained by the manual scanning process where blurred regions and stitching artifacts appear more often.

	few-shot *k*-means		classic training	
Accuracy	w/o aug.	w/ aug.	w/o aug.	w/ aug.
MIDI	**0.892**	0.873	0.890	0.890
M8	0.705	**0.874**	0.410	0.831
iSTIX	0.573	**0.718**	0.330	0.603
SCube	0.877	**0.882**	0.693	0.860
S360	0.679	**0.880**	0.353	0.867
S210	0.618	**0.869**	0.359	0.827
F1-score				
MIDI	**0.803**	0.751	0.798	0.785
M8	0.568	**0.745**	0.181	0.661
iSTIX	0.429	**0.546**	0.167	0.475
SCube	0.745	**0.762**	0.523	0.743
S360	0.533	**0.763**	0.094	0.739
S210	0.472	**0.746**	0.108	0.691

Table 2. Comparison of the accuracy and average F1-score of Multi-ProtoNets (*k=3* with adapted prototypes based on supports from a set of nine slides) and the classical approach, both with and without augmentation over all available scanners.

5. Discussion and Conclusion

Creating a robust and adaptive deep learning application is a challenging and relevant task in various fields. Few-shot learning and data augmentation have been powerful tools to enhance robustness and the possibility to quickly adapt to unseen data distributions. In this work we introduced Multi-ProtoNets, specifically a metric based approach that learns a multiple-cluster representation, and have shown its ability to adapt to distributional data shifts, based on only a few examples and without further training. We have experimentally demonstrated the positive effect of our multiple-cluster representation compared to a single-cluster representation, as given by Prototypical Networks, and a classically trained baseline for the task of tissue classification under shifted data. The evaluation was performed on six data sets recorded from different scanners with varying distributional similarity to the training data.

We extensively investigated the influence of the support set, used for prototype calculation at inference time, by a systematic variation in the number of new WSIs from the unseen scanners from which the support set was chosen for adaptation. As expected, a support set obtained from a larger number of slides provided more stable results. This is most likely due to the higher similarity between supports from the same WSI than between those from different WSIs. By introducing higher variance in the support set,

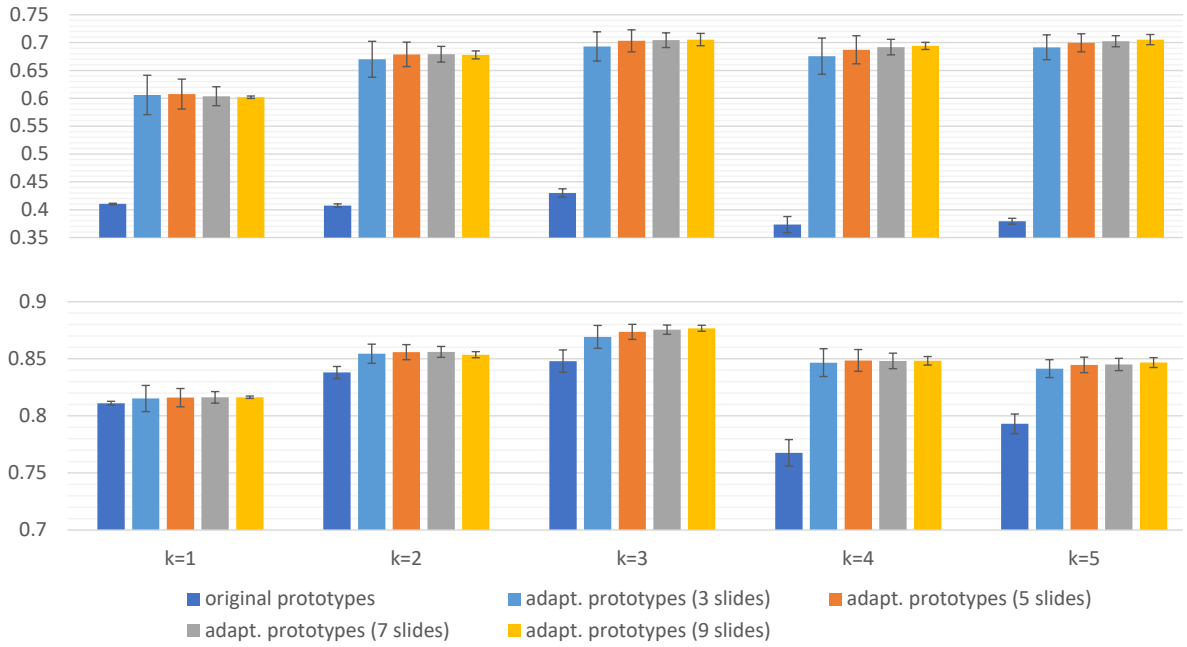

Figure 4. Accuracy of Multi-ProtoNets for different choices of k and numbers of slides for prototype calculation. The number of prototypes per class (k) is identical for training and testing. Note that $k = 1$ refers to Prototypical Networks. For testing the support set is either chosen from the original scanner ("original") or from a varying number of slides from the hold-out set of the specific scanners ("adapted"): M8 scanner (top) and SCube scanner (bottom). Note that we shifted the y-axis for better visualization.

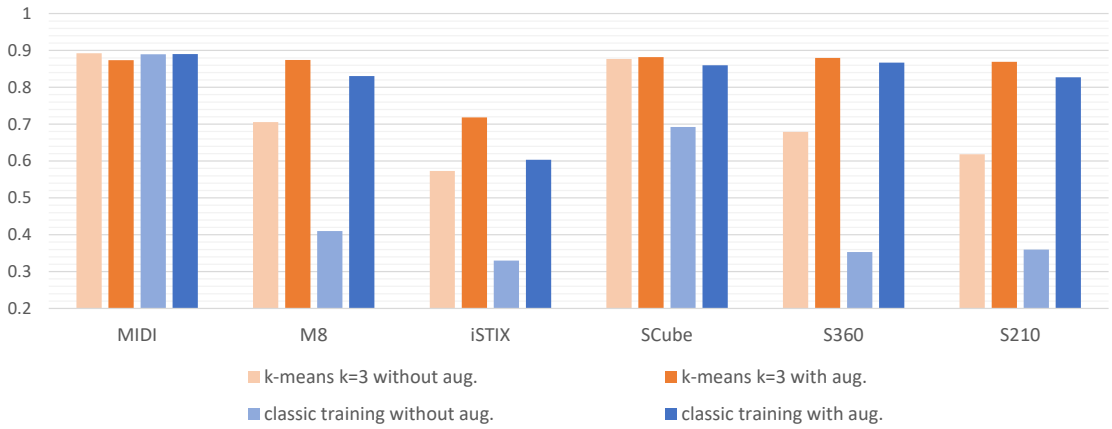

Figure 5. Accuracy of Multi-ProtoNets (using $k = 3$ prototypes per class with supports from nine slides) and of the classic baseline approach, each with and without augmentation during training over all available scanners.

as done by using different WSIs, more robust classification results were obtained. However, the differences in the results decreased more and more with increasing number of slides. This suggests that our model can be adapted to new scanners, also based on a smaller number of new slides and samples (*e.g.* five instead of nine slides) with only minimally lower accuracy.

The combination with domain-specific augmentation has further improved our results, as augmentation yields a more robust latent representation. We hypothesize that this latent space also allows for a more robust prototype selection.

As future work, we suggest experimenting with other, more complex backbone architectures to learn a more elaborate embedding. In addition, we want to investigate whether our conclusions are transferable on larger distributional shifts, for example the classification of tissue sections of other organs such as the bladder instead of the colon, which were additionally digitized by other scanners.

Acknowledgments

This work was supported by the Bavarian Ministry of Economic Affairs, Regional Development and Energy through the Center for Analytics – Data – Applications (ADA-Center) within the framework of "BAYERN DIGITAL II" (20-3410-2-9-8) and by the BMBF (16FMD01K, 16FMD02 and 16FMD03).

References

[1] Kelsey R Allen, Evan Shelhamer, Hanul Shin, and Joshua B Tenenbaum. Infinite mixture prototypes for few-shot learning. *arXiv preprint arXiv:1902.04552*, 2019. 2

[2] Antreas Antoniou, Harrison Edwards, and Amos Storkey. How to train your maml. *arXiv preprint arXiv:1810.09502*, 2018. 1

[3] Antreas Antoniou and Amos J Storkey. Learning to learn by self-critique. In *Advances in Neural Information Processing Systems*, pages 9940–9950, 2019. 1

[4] Aihua Cai, Wenxin Hu, and Jun Zheng. Few-shot learning for medical image classification. In Igor Farkaš, Paolo Masulli, and Stefan Wermter, editors, *Artificial Neural Networks and Machine Learning – ICANN 2020*, pages 441–452, Cham, 2020. Springer International Publishing. 5

[5] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. *arXiv preprint arXiv:1703.03400*, 2017. 1

[6] Aurélien Géron. *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems*. O'Reilly Media, 2019. 3

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 3

[8] Gabriel Huang, Hugo Larochelle, and Simon Lacoste-Julien. Are few-shot learning benchmarks too simple? solving them without task supervision at test-time. *arXiv e-prints*, pages arXiv–1902, 2019. 2

[9] Kian Kenyon-Dean, Andre Cianflone, Lucas Page-Caccia, Guillaume Rabusseau, Jackie Chi Kit Cheung, and Doina Precup. Clustering-oriented representation learning with attractive-repulsive loss. *arXiv preprint arXiv:1812.07627*, 2018. 4

[10] C. Khosla and B. S. Saini. Enhancing performance of deep learning models with different data augmentation techniques: A survey. In *2020 International Conference on Intelligent Engineering and Management (ICIEM)*, pages 79–85, 2020. 2

[11] Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov. Siamese neural networks for one-shot image recognition. In *ICML deep learning workshop*, volume 2. Lille, 2015. 2

[12] Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Irena Gao, et al. Wilds: A benchmark of in-the-wild distribution shifts. *arXiv preprint arXiv:2012.07421*, 2020. 1

[13] Petr Kuritcyn, Carol I Geppert, Markus Eckstein, Arndt Hartmann, Thomas Wittenberg, Jakob Dexl, Serop Baghdadlian, David Hartmann, Dominik Perrin, Volker Bruns, et al. Robust slide cartography in colon cancer histology: Evaluation on a multi-scanner database. In *Bildverarbeitung für die Medizin 2021: Proceedings, German Workshop on Medical Image Computing, Regensburg, March 7-9, 2021*, pages 229–234. Springer Fachmedien Wiesbaden, 2021. 1, 2, 5, 6

[14] Xiaoxu Li, Tao Tian, Yuxin Liu, Hong Yu, Jie Cao, and Zhanyu Ma. Adaptive multi-prototype relation network. In *2020 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pages 1707–1712. IEEE, 2020. 2

[15] Thomas Mensink, Jakob Verbeek, Florent Perronnin, and Gabriela Csurka. Distance-Based Image Classification: Generalizing to new classes at near-zero cost. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(11):2624–2637, Nov. 2013. 2

[16] Viraj Prabhu, Anitha Kannan, Murali Ravuri, Manish Chaplain, David Sontag, and Xavier Amatriain. Few-shot learning for dermatological disease diagnosis. In Finale Doshi-Velez, Jim Fackler, Ken Jung, David Kale, Rajesh Ranganath, Byron Wallace, and Jenna Wiens, editors, *Proceedings of the 4th Machine Learning for Healthcare Conference*, volume 106 of *Proceedings of Machine Learning Research*, pages 532–552, Ann Arbor, Michigan, 09–10 Aug 2019. PMLR. 3

[17] Mengye Ren, Eleni Triantafillou, Sachin Ravi, Jake Snell, Kevin Swersky, Joshua B Tenenbaum, Hugo Larochelle, and Richard S Zemel. Meta-learning for semi-supervised few-shot classification. *arXiv preprint arXiv:1803.00676*, 2018. 2

[18] Oren Rippel, Manohar Paluri, Piotr Dollar, and Lubomir Bourdev. Metric learning with adaptive density discrimination. *arXiv preprint arXiv:1511.05939*, 2015. 2

[19] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *Advances in neural information processing systems*, pages 4077–4087, 2017. 1, 2

[20] Chetan L. Srinidhi, Ozan Ciga, and Anne L. Martel. Deep neural network models for computational histopathology: A survey. *Medical Image Analysis*, 67:101813, 2021. 1

[21] Qianru Sun, Yaoyao Liu, Tat-Seng Chua, and Bernt Schiele. Meta-transfer learning for few-shot learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 403–412, 2019. 1

[22] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. Learning to compare: Relation network for few-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1199–1208, 2018. 2

[23] David Tellez, Maschenka Balkenhol, Nico Karssemeijer, Geert Litjens, Jeroen van der Laak, and Francesco Ciompi. H and E stain augmentation improves generalization of convolutional networks for histopathological mitosis detection.

In *Medical Imaging 2018: Digital Pathology*, volume 10581, page 105810Z. International Society for Optics and Photonics, 2018. 2, 5

[24] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, koray kavukcuoglu, and Daan Wierstra. Matching networks for one shot learning. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016. 2

[25] Yaqing Wang, Quanming Yao, James T Kwok, and Lionel M Ni. Generalizing from a few examples: A survey on few-shot learning. *ACM Computing Surveys (CSUR)*, 53(3):1–34, 2020. 2

[26] Jaesik Yoon, Taesup Kim, Ousmane Dia, Sungwoong Kim, Yoshua Bengio, and Sungjin Ahn. Bayesian model-agnostic meta-learning. *Advances in Neural Information Processing Systems*, 31:7332–7342, 2018. 1

[27] Amy Zhao, Guha Balakrishnan, Fredo Durand, John V Guttag, and Adrian V Dalca. Data augmentation using learned transformations for one-shot medical image segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8543–8553, 2019. 2