A Comparative Study of Vision Transformer Encoders and Few-shot Learning for Medical Image Classification

Maxat Nurgazin Nguyen Anh Tu*
Department of Computer Science, School of Engineering and Digital Sciences
Nazarbayev University, 53 Kabanbay Batyr Ave., Astana, Kazakhstan, 010000
{maxat.nurgazin, tu.nguyen}@nu.edu.kz

Abstract

Recently, computer vision has been significantly impacted by Vision Transformer (ViT) networks. These deep models have also succeeded in medical image classification. However, most existing deep learning-based methods primarily rely on a lot of labeled data to train reliable classifiers for accurate prediction. This requirement might be impractical in the medical field, where the data is limited and manual annotation is expensive. Therefore, this study explores the application of ViT in few-shot learning scenarios for medical image analysis, addressing the challenges posed by limited data availability. We evaluate various ViT models alongside few-shot learning algorithms (i.e., ProtoNet, MatchingNet, and Reptile), perform cross-domain experiments, and analyze the impact of data augmentation techniques. Our findings indicate that when combined with ProtoNets, ViT architectures outperform CNN-based counterparts and achieve competitive performance against state-of-the-art approaches on benchmark datasets. Cross-domain experiments further reveal the effectiveness of ViT models in few-shot medical image classification.

1. Introduction

Medical image analysis (MIA) is crucial in diagnosing various diseases and conditions, underpinning its significance in healthcare. The sheer amount of data generated requires creating effective and precise automated MIA techniques. Recently, machine learning, particularly deep learning, has shown itself as a viable approach to tackle this challenge, specifically in the context of medical image classification (MIC), the central theme of this study. Convolutional Neural Networks (CNNs) have set the benchmark in numerous medical imaging applications. CNNs operate based on a localized convolution operation that ensures translational equivariance, enabling the extraction of local spatial features that can be aggregated to form higher-order representations. However, their capacity to learn long-range pixel relationships is limited. The Transformer architecture introduced by [26] has addressed this limitation and later adapted to computer vision as "Vision Transformer" (ViT) [8]. Unlike conventional CNNs, ViTs leverage a self-attention mechanism for understanding the overall representation of input images, learning both short-range and long-range input relationships. This ability makes them particularly effective for large and complex medical images. Transformers have even been shown to surpass CNNs when employed with larger datasets or self-supervised learning for medical imaging tasks [16]. Furthermore, Transformers incorporate saliency maps that facilitate an understanding of model decisions, enabling experts to validate model outcomes. Because of the distinct advantages of ViTs, the research community has made a great attempt to adapt them for medical imaging applications [10].

Classical supervised deep-learning methods are data-hungry and perform well when large annotated datasets are available. However, they may not be practical or successful when data is limited or manual annotation is costly, which is often the case in many medical imaging subfields. Few-shot learning (FSL) [2], a technique that trains a model to recognize and classify new objects or concepts with minimal examples, has recently emerged as a potential solution to the problem of limited labeled data in MIC. FSL aims to mimic human learning, often involving grasping new concepts from a few samples. For example, a computer vision model trained with FSL approaches can accurately identify rare diseases after training with a few medical images. As shown in [24], the feature extractor or encoder learned from the good neural architecture can produce robust representations, enabling fast adaption to new tasks at test times. Moreover, Hu et al. [13] demonstrated that an FSL pipeline based on the ViT encoder could deliver impressive results on standard FSL benchmarks. To the best of our knowledge, this approach has not been applied to MIC; hence, it is unknown whether the few-shot MIC can benefit ViT sim-

*Corresponding author.

ilarly. Therefore, our study aims to explore the robustness of ViTs in an FSL setting for MIC.

In this paper, we will primarily investigate the application of ViTs in an FSL scenario for MIC and draw comparisons with traditional CNNs. For this purpose, we employ prominent pre-trained ViT models alongside FSL algorithms, such as Prototypical Networks [21], Matching Networks [27] and Reptile [18], and compare their performance with the CNN-based counterparts. The datasets used include ISIC 2018 [30], BreakHis [23], and Pap Smear [14]. Furthermore, another approach to address the scarcity of labeled data is data augmentation, which involves creating additional training data by modifying existing data. Accordingly, this research will delve into the influence of augmentation techniques (e.g., Cutout [7], Mixup [29], and Cutmix [28]) on the effectiveness of ViT for FSL. Similar to [2], we will also evaluate the effects of domain shift between base and novel classes in natural-to-medical and medical-to-medical cross-domain experiments. This evaluation setting is particularly practical since the domains of the source (e.g., natural images) and target (medical images) are generally different. Therefore, understanding the effect of domain shift can allow us to better assess the generalization performance of different few-shot learners.

Our major contributions can be summarized as follows:

1. We investigate the efficacy of various ViT models for few-shot medical image classification.
2. We study how different few-shot learning algorithms impact the performance of ViT models.
3. We analyze the impact of advanced data augmentation techniques on ViT models.
4. We explore the effect of a cross-domain scenario on the performance of few-shot learners.
5. Our methods achieve state-of-the-art performance on challenging medical datasets of few-shot medical image classification.

2. Related Work

This section reviews the literature on few-shot learning with ViTs and the application of few-shot learning in medical image classification.

2.1. Few-shot Learning with ViT

This section discusses recent research papers on few-shot learning utilizing the ViT architecture. Comprehensive reviews on the current state of few-shot learning can be found in several survey papers [22, 12].

Limited research has been conducted on applying ViTs in FSL scenarios. A noteworthy example is the work of Hu et al. [13], which compared a simple FSL pipeline with advanced algorithms. Their approach employed ViT small and ResNet50 as backbone models and outperformed the state-of-the-art, particularly when using the Transformer backbone. Chen et al. [4] proposed an architecture that uses image masking for few-shot learning, demonstrating superior performance over a standard ViT. Our research extends these works by applying a simple ViT-based pipeline to few-shot medical image classification.

2.2. Medical Image Classification and FSL

Singh et al. [20] presented MetaMed, a meta-learning-based approach for few-shot learning in medical image classification, which significantly outperformed transfer learning. Dai et al. [6] proposed PFEMed, a novel few-shot classification method for medical images that utilized a dual-encoder structure. Cherti and Jitsev [5] explored the effects of the pre-training scale in both in-domain and out-of-domain transfer settings. These works highlight the potential of few-shot learning approaches in medical image classification, despite the challenges posed by the scarcity and quality of annotated medical images.

3. Methodology

This section defines the problem of few-shot medical image classification, outlines the overall system pipeline, and describes the methodology.

3.1. Problem Definition

Consider a collection $D = \{D_1, D_2, ..., D_n\}$ of n medical datasets, where each dataset D_k includes pairs $(\mathbf{x}, y)_j$, representing an image and its corresponding label (ground-truth). Each dataset is divided into a meta-test set ($D_{meta-test}$), comprising classes with fewer representative images (rare diseases), and a meta-train set ($D_{meta-train}$), which includes the remaining classes. The strategy is to leverage the extensive data available in $D_{meta-train}$ (base class data). When using initialization-based methods like Reptile [18], the objective is to learn better initial weights and then fine-tune the model with limited data (novel class data). For metric learning-based methods like MatchingNet [27] and ProtoNet [21], the aim is to develop a model that creates an effective embedding space where the feature representation of a query is near support features or prototypes of its corresponding class and far from support features or prototypes of other classes, respectively. This approach enables easy identification of similar items. The overall system pipeline is shown in Figure 1, where ProtoNet, MatchingNet, and Reptile function as support set conditioned models. The figure's bottom part illustrates the architectures of these models, where f_θ denotes the ViT encoder pre-trained on ImageNet1K (θ is the ViT network parameter).

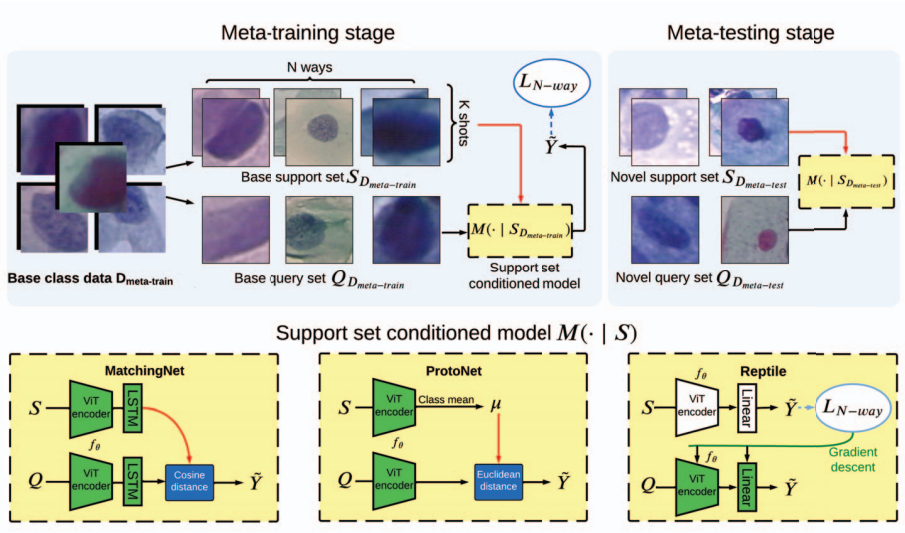

Figure 1. **Overall System Pipeline.**

3.2. Meta-learning

Few-shot learning aims to build machine learning models that can efficiently generalize to new tasks, given only a few labeled examples from each class in the target domain. Few-shot learning tasks' complexity can be described as N-way-K-shot, where N denotes the number of classes, and K represents the number of samples from each class used for training. Multiple approaches exist for few-shot learning, including a meta-learning perspective. In this approach, the model learns to solve new few-shot tasks by gaining experience from solving other tasks, divided into meta-training and meta-testing phases. Data is presented episodically in each phase, with the support set acting as the training set and the query set as the test set. Transfer learning is another few-shot learning approach, where the model is pre-trained on a large dataset and then fine-tuned on the limited support set. However, this approach is less effective when there is a significant domain gap between the source and target datasets. Data augmentation is yet another technique for addressing few-shot learning, where new samples are created by augmenting samples from a limited support set.

3.2.1 ViT Encoder

The ViT architecture was introduced in [8] as an adaptation of the original Transformer model to the field of Computer Vision. It processes an input image by dividing it into non-overlapping patches and creating the linear embedding from these patches based on the linear projection. Positional encoding is added to this linear embedding to incorporate the positional information. Later, these embeddings are fed into a Transformer encoder. The architecture of a ViT is depicted in Fig. 2. In this work, we use the ViT encoder as a feature extractor (f_θ) mapping the image to a D-dimensional space.

The encoder uses Scaled Dot-Product Attention (SDPA), which calculates attention scores between pairs of input tokens, which are then used to compute a weighted average of the input embeddings. Specifically, query ($\mathbf{Q}$), key ($\mathbf{K}$), and value ($\mathbf{V}$) matrices are computed by multiplying the input embeddings ($\mathbf{X} \in \mathbb{R}^{N \times D}$) with learnable weight matrices $\mathbf{W}^Q, \mathbf{W}^K,$, and $\mathbf{W}^V$, i.e.,$\mathbf{Q} = \mathbf{X}\mathbf{W}^Q, \mathbf{K} = \mathbf{X}\mathbf{W}^K, \mathbf{V} = \mathbf{X}\mathbf{W}^V$. Then, SDPA is calculated as follows:

$$SDPA(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = softmax(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}})\mathbf{V} \qquad (1)$$

To capture diverse dependencies and learn different aspects of the input, ViT employs multiple such SDPA heads stacked together. Each head has its own set of learnable weight matrices, allowing the model to simultaneously attend to different parts of the input sequence. This mechanism is called Multi-Head Self-Attention (MHSA) in the ViT architecture. It allows the model to capture both local and global dependencies and relationships between different patches in the image, facilitating the learning of rich and informative representations from images.

To use a ViT model as a backbone of the ProtoNet or MatchingNet algorithm, we remove the MLP head of the model and use it as a pure feature extractor. For Reptile, we replace the existing head with a new head that has an output size of N (from N-way-K-shot task).

3.2.2 Prototypical Networks and Matching Networks

Prototypical Networks (ProtoNet [21]) aim to learn a prototype for each class in the embedding space. Given a set

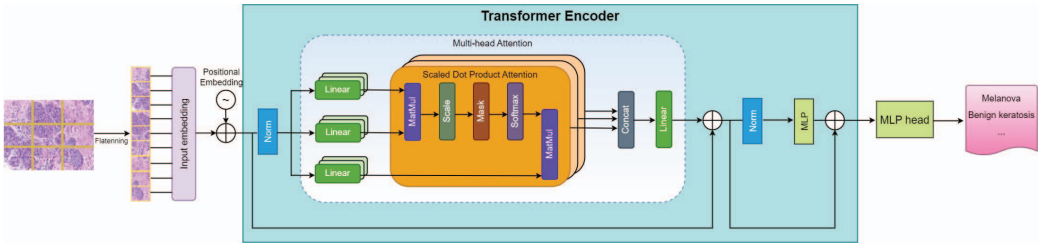

Figure 2. **ViT model.**

of support samples, the model learns a function f_θ to embed the images into features. The feature embedding corresponds to the ViT backbone we presented above.

The prototype μ_k for class k is computed as the mean of the embedded support samples belonging to that class:

$$\mu_k = \frac{1}{N_k} \sum_{(\mathbf{x}_i, y_i) \in \mathbf{S}, y_i = k} f_\theta(\mathbf{x}_i) \tag{2}$$

Here, $\mathbf{S}$ represents the support set, and N_k is the number of support samples in class k.

For a given query $\mathbf{x}$, the model computes its embedding and determines the class by finding the prototype with the smallest Euclidean distance. Specifically, the probability of the query $\mathbf{x}$ belonging to class k is computed as follows:

$$p(y = k|\mathbf{x}) = \frac{exp\left(-\|f_\theta(\mathbf{x}) - \mu_k\|^2\right)}{\sum_{k'} exp\left(-\|f_\theta(\mathbf{x}) - \mu_{k'}\|^2\right)} \tag{3}$$

Matching Networks (MatchingNet) [27] is similar to ProtoNet, except it computes an average cosine distance for each class by comparing the feature representation of a query with each feature representation from a support set. Additionally, it uses LSTM for full context embedding from the whole support set.

3.2.3 Reptile

Reptile [18] is a type of initialization-based meta-learning and operates by iteratively updating the model's weights through a two-level process: inner loop updates and outer loop updates. The inner loop focuses on learning from individual tasks, while the outer loop learns across tasks. The simplicity of Reptile allows for faster training and easier implementation compared to MAML [9], as it does not require computation of second-order gradients. Cross entropy loss was used to update the model's weights in the meta-training and meta-testing phases. For a task T_i, it is given by

$$\mathcal{L} = - \sum_{\mathbf{x}_i, y_i \sim T_i} y_i log(\phi f_\theta(\mathbf{x}_i) + (1 - y_i) log(1 - \phi f_\theta(\mathbf{x}_i)) \tag{4}$$

where ϕ is the parameter of a linear classifier trained on $D_{meta-train}$ and used as the predictor on $D_{meta-test}$

3.3. Data Augmentation Techniques

In deep learning, data augmentation is crucial in achieving high performance, mainly when working with small datasets. This importance is amplified in FSL scenarios, where the available data is limited. Augmentation techniques serve as regularizers that encourage models to learn more generalized representations, as they need to predict correct labels based on augmented inputs, effectively reducing overfitting. These techniques (i.e., Cutout [7], Mixup [29], and Cutmix [28]) were employed in [20] and significantly improved FSL accuracy. However, given that larger models are used in our research, assessing if these techniques yield similarly promising results is essential. Note that only Cutout is compatible with the ProtoNet while the other two techniques are not applicable due to modifying the support set labels. By applying these augmentation techniques, we aim to improve the performance of ViTs in few-shot learning scenarios and assess their effectiveness compared to traditional methods.

4. Experiments

4.1. Dataset Description

Three publicly available medical imaging datasets were chosen for this study. Each dataset contains at least six classes, allowing for both 2-way and 3-way n-shot learning.

BreakHis) The BreakHis dataset [23] consists of 9,109 microscopic images of breast tumor tissues collected from 82 patients and captured at magnification levels of 40, 100, 200, and 400. The dataset is divided into eight classes, with five classes selected as meta-train classes and the remaining classes designated as meta-test classes.

ISIC 2018) The ISIC 2018 Skin Lesion dataset [30] comprises 10,015 dermoscopic images spanning seven classes. The distribution of diseases in the dataset reflects real-world prevalence, with more images for benign lesions than malignant ones. Four classes with the most samples were selected as meta-train classes, while the remaining three were designated for meta-testing.

Pap Smear) The Pap Smear dataset [14] consists of microscopic images of cervical smears taken at Herlev University Hospital. The dataset contains 917 images, unevenly

distributed across seven distinct classes. Four classes with the most samples were selected as meta-train classes, while the remaining three classes were selected for meta-testing.

4.2. Models

This section discusses various models used in the experiments. These tested models can be grouped into three categories with the number of parameters for each model:

- Standard ViT: Three models from the ViT family [8] were utilized, namely ViT_tiny(5.5M), ViT_small(22M), and ViT_base(85M).
- Other ViT variants: The Mobile ViT (1.4M),i.e., MViT [17], DeiT_base(85M) [25], and Swin_base(86M) [15] models were selected to investigate the performance of alternative ViT architectures.
- CNN models: The ResNet50 [11] and VGG16 [19] models were included for comparison with the ViT models. ResNet50 has 23.5M parameters, while VGG16 has 134M parameters.

All models were pre-trained on ImageNet1K, a widely used dataset for training computer vision models.

4.3. Implementation Details

The implementation was done in PyTorch. Pre-trained model checkpoints were obtained from Timm library [1].

Pre-trained model checkpoints were utilized during the training phase, and data augmentation techniques were employed to improve generalization. It is worth noting that the episodic task sampling in FSL helps mitigate the effects of class imbalance, as few-shot learners see an equal number of samples from each class.

For ProtoNet and MatchingNet, the model was trained for 20 epochs, as further training epochs resulted in overfitting. Each epoch consisted of 500 episodes or tasks. The stochastic gradient descent (SGD) optimizer was used with a learning rate of 10^{-5} or 10^{-6} on the model, and a momentum of 0.9, depending on the dataset. A cosine annealing learning rate schedule was also employed.

For Reptile, the SGD optimizer was used with a learning rate of 10^{-3} for the inner optimization problem and SGD with a learning rate (step size) of 10^{-1} for the outer meta-update step. The backbone was trained for 100 meta-iterations with a batch size of 10 tasks during meta-training. The batch size was set to 10 tasks per meta-iteration in both training and testing. The inner problem was experimented with 50 adaptation steps on each task.

Evaluation Protocol. Accuracy (%) was used as the evaluation metric, a common performance indicator for few-shot classification tasks. To assess performance on the BreakHis, ISIC 2018, and Pap Smear datasets, 400 episodes from the novel classes in the test set were randomly selected each time, and the average accuracy rate for image classification was computed.

Table 1. Transfer learning results on ISIC 2018

Setting	Model	2-way			3-way		
		3-shot	5-shot	10-shot	3-shot	5-shot	10-shot
Baseline	ViT_small	75.50	81.88	85.93	61.77	66.78	74.48
	ResNet50	70.63	72.18	73.88	52.73	57.35	60.67
Baseline-PN	ViT_small	**81.08**	**83.59**	**88.30**	**68.88**	**73.68**	**79.27**
	ResNet50	73.95	77.86	81.00	59.23	63.05	68.33

Table 2. Transfer learning results on BreakHis

Mag	Meta-learning	Model	2-way			3-way		
			3-shot	5-shot	10-shot	3-shot	5-shot	10-shot
40	Baseline	ViT_small	78.80	**84.18**	**88.13**	68.15	74.70	81.05
		ResNet50	76.48	82.33	84.88	65.35	69.68	74.37
	Baseline-PN	ViT_small	**82.41**	84.08	87.83	**72.19**	**76.96**	**82.43**
		ResNet50	69.08	74.49	79.22	58.57	63.30	68.53
100	Baseline	ViT_small	73.23	78.98	84.13	61.32	67.67	75.25
		ResNet50	75.10	**81.83**	84.10	63.82	69.72	71.92
	Baseline-PN	ViT_small	**76.29**	81.09	**84.24**	**68.07**	**72.93**	**77.83**
		ResNet50	74.19	78.79	82.61	63.32	68.55	73.40
200	Baseline	ViT_small	70.45	74.80	81.88	58.87	66.42	73.02
		ResNet50	67.08	72.90	77.35	55.18	61.57	65.72
	Baseline-PN	ViT_small	**72.39**	**78.11**	**82.36**	**61.93**	**69.03**	**73.99**
		ResNet50	62.56	66.74	73.40	51.39	55.47	60.77
400	Baseline	ViT_small	72.93	78.43	82.50	62.75	67.75	**72.98**
		ResNet50	68.00	72.43	76.23	54.02	59.77	63.88
	Baseline-PN	ViT_small	**75.00**	**79.19**	**83.28**	**65.12**	**69.50**	72.66
		ResNet50	62.71	65.76	68.64	48.59	53.02	58.31

Table 3. Transfer learning results on Pap Smear

Meta-learning	Model	2-way			3-way		
		3-shot	5-shot	10-shot	3-shot	5-shot	10-shot
Baseline	ViT_small	86.43	90.95	93.06	78.13	83.69	87.32
	ResNet50	77.58	81.05	86.28	66.12	69.83	75.17
Baseline-PN	ViT_small	**92.58**	**94.17**	**95.85**	**85.10**	**88.09**	**91.39**
	ResNet50	81.83	84.58	86.71	70.70	73.97	77.30

4.4. Result Analysis

4.4.1 Transfer Learning Results

This section provides the results for transfer learning, which is used as our baseline. We have 2 different transfer learning settings, which all start with a pre-trained model checkpoint: (1) "Baseline" - fine-tune the model on base classes in a supervised manner and perform 50 adaptation steps for tasks from novel classes during the evaluation phase, (2) "Baseline-PN" - fine-tune the model on base classes in a supervised manner and perform inference using the ProtoNet algorithm without performing meta-training. These results are summarized in Tables 1-3.

From transfer learning results, it can be observed that a fine-tuned ViT_small outperforms ResNet50 in all tests except for the 2-way-5-shot task on BreakHis X100 dataset. Regarding the comparison between settings, ViT_small in the Baseline-PN demonstrated higher scores in the majority of cases when compared with the Baseline.

Table 4. Performance of models using different meta-learning algorithms for ISIC 2018 dataset.

Algorithm	Model	2-way			3-way		
		3-shot	5-shot	10-shot	3-shot	5-shot	10-shot
ProtoNet	MViT	74.64	76.94	81.50	60.60	64.23	69.23
	ViT_tiny	81.03	83.61	86.52	67.84	71.82	77.68
	ViT_small	**84.35**	**86.70**	89.72	72.10	76.18	81.45
	ViT_base	83.94	86.02	**90.26**	**72.75**	**77.69**	**81.99**
	DeIT_base	72.17	76.53	81.40	57.86	62.38	69.07
	Swin_base	82.49	84.17	89.12	70.75	74.67	79.92
	ResNet50	66.62	68.65	72.81	51.43	53.83	58.34
	VGG16	74.11	78.17	82.11	60.68	64.58	70.84
Reptile	MViT	62.80	67.00	71.80	53.00	54.47	60.33
	ViT_tiny	75.80	78.40	83.50	64.13	68.67	75.13
	ViT_small	70.30	76.10	80.40	63.13	72.13	78.53
	ViT_base	59.30	67.40	72.70	53.27	62.27	70.53
	DeIT_base	72.80	79.40	83.00	61.73	64.73	73.60
	Swin_base	67.30	74.20	81.10	60.60	69.00	75.93
	ResNet50	71.70	72.70	76.50	47.60	51.60	54.93
	VGG16	64.70	72.90	78.60	56.53	62.40	70.67
MatchingNet	MViT	72.41	75.70	78.59	58.79	62.00	65.62
	ViT_tiny	76.66	79.88	83.41	63.42	66.62	71.95
	ViT_small	78.40	81.61	86.34	65.50	70.00	76.47
	ViT_base	79.81	84.19	88.21	67.70	73.30	79.17
	DeIT_base	73.67	77.11	81.60	58.23	62.54	70.47
	Swin_base	72.84	75.60	80.12	58.66	63.09	68.34
	ResNet50	67.99	71.69	75.66	52.93	56.53	61.60
	VGG16	72.20	75.94	79.90	59.76	61.63	67.98

Table 5. Performance of models using different meta-learning algorithms for BreakHis with X40 magnification dataset.

Algorithm	Model	2-way			3-way		
		3-shot	5-shot	10-shot	3-shot	5-shot	10-shot
Protonet	MViT	76.19	80.92	84.51	64.38	69.84	76.57
	ViT_tiny	69.91	73.42	77.40	56.58	62.02	66.73
	ViT_small	77.18	81.79	84.80	69.69	73.44	76.63
	ViT_base	78.05	81.59	85.21	68.54	74.29	79.61
	DeIT_base	72.80	77.71	82.81	60.88	67.16	73.37
	Swin_base	**81.16**	**85.78**	90.00	**73.01**	**78.98**	83.79
	ResNet50	70.03	73.76	77.48	57.37	62.98	67.58
	VGG16	77.10	81.62	84.97	66.66	72.19	77.51
Reptile	MViT	72.10	78.50	81.20	53.40	55.53	61.27
	ViT_tiny	63.50	69.90	81.50	53.13	64.80	73.40
	ViT_small	69.90	79.20	84.80	57.47	64.47	72.73
	ViT_base	65.10	67.90	67.20	47.53	56.20	63.33
	DeIT_base	74.40	81.80	87.90	54.00	67.67	76.13
	Swin_base	68.90	71.00	81.70	53.13	57.87	73.07
	ResNet50	67.50	71.00	76.20	56.47	63.67	64.13
	VGG16	74.80	77.10	84.30	61.13	71.67	79.13
MatchingNet	MViT	74.92	80.70	85.10	64.79	70.76	78.16
	ViT_tiny	68.86	74.31	80.86	55.31	62.21	71.71
	ViT_small	77.81	82.34	89.45	67.06	74.23	82.10
	ViT_base	80.66	84.66	**90.10**	71.27	78.77	**86.22**
	DeIT_base	73.31	78.69	83.93	61.94	68.07	76.20
	Swin_base	77.92	83.61	88.74	68.20	74.77	81.52
	ResNet50	71.30	73.98	78.20	57.84	62.93	69.57
	VGG16	76.51	81.66	86.46	65.99	71.37	79.08

Table 6. Performance of models using different meta-learning algorithms for BreakHis with X100 magnification dataset.

Algorithm	Model	2-way			3-way		
		3-shot	5-shot	10-shot	3-shot	5-shot	10-shot
Protonet	MViT	76.89	79.60	84.65	64.51	71.43	77.05
	ViT_tiny	75.34	79.44	83.53	62.64	69.88	75.18
	ViT_small	**80.64**	**83.80**	87.62	**69.39**	**75.91**	81.47
	ViT_base	79.33	81.65	84.62	68.52	73.27	76.38
	DeIT_base	68.91	73.81	78.16	56.38	62.25	67.54
	Swin_base	79.46	82.86	86.26	68.34	74.28	80.51
	ResNet50	68.62	72.12	73.31	55.80	60.28	61.88
	VGG16	74.59	78.50	81.84	61.21	65.83	72.16
Reptile	MViT	73.45	80.90	83.70	55.17	60.23	66.57
	ViT_tiny	66.60	72.50	78.10	54.13	61.80	71.70
	ViT_small	68.10	75.60	81.60	54.40	63.13	72.20
	ViT_base	56.25	60.20	66.45	40.63	46.40	55.67
	DeIT_base	67.40	71.20	79.05	47.53	51.03	62.50
	Swin_base	66.70	72.60	80.00	54.47	59.63	74.40
	ResNet50	71.00	76.90	78.30	53.87	58.60	60.80
	VGG16	66.00	70.90	79.90	54.33	65.40	75.67
MatchingNet	MViT	78.57	82.78	88.61	67.62	74.79	81.03
	ViT_tiny	71.85	75.71	84.16	59.51	67.17	74.78
	ViT_small	76.53	82.09	88.33	67.13	72.88	81.80
	ViT_base	76.54	82.27	88.39	66.04	73.31	81.83
	DeIT_base	69.76	72.86	79.76	55.33	60.59	67.17
	Swin_base	77.68	83.24	**89.60**	67.90	74.61	**82.88**
	ResNet50	73.45	76.58	79.14	59.53	62.70	66.98
	VGG16	74.84	78.12	82.29	61.79	66.20	73.38

Table 7. Performance of models using different meta-learning algorithms for BreakHis with X200 magnification dataset.

Algorithm	Model	2-way			3-way		
		3-shot	5-shot	10-shot	3-shot	5-shot	10-shot
Protonet	MViT	71.17	75.50	80.92	59.00	64.80	70.63
	ViT_tiny	65.83	69.62	73.21	52.42	56.14	61.47
	ViT_small	**75.10**	**79.40**	83.33	**64.16**	69.12	75.38
	ViT_base	72.20	76.91	81.94	60.54	66.23	71.98
	DeIT_base	68.77	74.69	77.33	55.33	60.93	65.48
	Swin_base	66.38	67.12	72.69	50.84	54.73	60.52
	ResNet50	69.47	72.80	76.30	55.57	60.59	65.37
	VGG16	68.12	73.52	78.35	55.24	61.10	66.48
Reptile	MViT	69.90	76.00	79.10	49.67	52.20	60.13
	ViT_tiny	59.50	65.70	70.70	47.80	53.13	60.80
	ViT_small	62.70	71.00	80.80	47.27	53.80	60.73
	ViT_base	57.00	57.30	62.00	39.93	42.47	47.20
	DeIT_base	72.80	75.10	80.20	49.47	53.07	64.87
	Swin_base	61.70	68.20	73.20	48.20	53.07	66.53
	ResNet50	63.90	69.90	72.40	50.93	54.00	60.07
	VGG16	64.50	74.70	83.50	50.27	60.80	70.13
MatchingNet	MViT	69.89	73.30	80.40	56.55	63.20	70.01
	ViT_tiny	65.50	71.70	78.74	53.64	59.58	68.14
	ViT_small	71.83	77.33	84.47	59.64	66.91	75.92
	ViT_base	73.05	79.36	**87.35**	62.42	**70.51**	**79.27**
	DeIT_base	69.14	74.05	78.94	55.19	61.01	67.53
	Swin_base	68.54	73.41	80.71	54.77	60.79	70.50
	ResNet50	70.23	73.14	77.92	55.62	60.45	66.75
	VGG16	70.30	74.49	81.10	57.07	62.96	70.54

4.4.2 Meta-Training Results

This section delves into analyzing the performance of few-shot classification models utilizing ProtoNet, MatchingNet and Reptile meta-learning algorithms. The findings are presented in Tables 4 through 9, where the highest scores within the algorithm are underlined, and the highest scores across all algorithms are given in bold. By comparing these results with baselines, it can be observed that the best ViTs in conjunction with ProtoNet or MatchingNet demonstrated substantial performance improvements across most tasks. This shows the effectiveness of performing meta-training with ProtoNet or MatchingNet. Inferior results were detected in 1 task of BreakHis X40, 4 tasks of BreakHis X400, and 4 tasks of Pap smear datasets. ViT_small paired with the ProtoNet showed the highest average accuracy across all datasets outperforming larger models. Generally, ViT_small and ViT_base have the highest scores when paired with either of ProtoNet or MatchingNet. The former seems a more favorable choice considering models' size difference (22M parameters against 85M). Regarding the performance of CNNs, when used as a backbone of ProtoNet or MatchingNet, they yielded inferior results compared to ViT counterparts, especially ResNet50. This pattern coincides with the findings reported by [3], where ResNet50's scores dipped significantly after meta-training with ProtoNet.

As for the performance of models with the Reptile algorithm, we can observe that it is significantly lower when compared with other algorithms, usually not beating the baselines. Generally, its performance highly depends on hyperparameters, especially the number of inner adaptation steps. Typically, the accuracy increases as the number of inner adaptations steps increases. However, processing times increase linearly with this number. Considering the simplicity of use and training, superior performance across

Table 8. Performance of models using different meta-learning algorithms for BreakHis with X400 magnification dataset.

Algorithm	Model	2-way			3-way		
		3-shot	5-shot	10-shot	3-shot	5-shot	10-shot
Protonet	MViT	68.80	73.86	78.12	56.37	61.88	67.61
	ViT_tiny	65.71	69.41	73.45	52.54	56.93	62.10
	ViT_small	70.45	74.60	80.66	57.99	63.34	69.08
	ViT_base	71.04	75.01	80.62	57.49	62.86	66.81
	DeIT_base	67.17	70.8	76.48	54.27	59.25	64.04
	Swin_base	67.01	71.51	74.58	52.47	56.73	63.00
	ResNet50	66.59	68.90	71.74	51.59	54.24	58.50
	VGG16	67.20	70.86	75.99	54.16	58.20	63.99
Reptile	MViT	67.40	71.00	80.60	55.27	58.47	61.67
	ViT_tiny	57.40	53.90	56.90	49.07	54.60	66.80
	ViT_small	63.00	68.90	81.90	47.13	52.27	58.40
	ViT_base	60.40	62.70	63.10	42.40	45.40	48.73
	DeIT_base	61.30	70.60	78.70	50.13	56.80	67.27
	Swin_base	64.00	70.80	79.00	50.20	55.73	64.47
	ResNet50	63.80	67.30	72.10	46.40	51.67	53.73
	VGG16	67.60	73.70	82.80	51.07	55.80	62.20
MatchingNet	MViT	68.88	73.76	79.74	55.53	62.15	68.97
	ViT_tiny	66.47	71.90	77.28	55.16	60.61	68.25
	ViT_small	**71.66**	76.35	82.33	**58.14**	64.28	72.88
	ViT_base	69.85	**76.44**	**83.64**	57.63	**65.35**	**76.06**
	DeIT_base	67.56	71.62	79.88	54.91	61.50	68.31
	Swin_base	69.10	74.79	82.47	55.80	62.20	71.18
	ResNet50	67.65	69.71	73.90	52.32	55.13	59.41
	VGG16	66.05	71.76	78.65	54.93	59.82	67.60

Table 9. Performance of models using different meta-learning algorithms for Pap Smear dataset.

Algorithm	Model	2-way			3-way		
		3-shot	5-shot	10-shot	3-shot	5-shot	10-shot
Protonet	MViT	80.84	84.36	86.88	68.04	73.24	78.37
	ViT_tiny	84.65	86.96	88.86	74.33	77.92	81.17
	ViT_small	**92.40**	**94.05**	**94.90**	**86.38**	**89.09**	**90.62**
	ViT_base	92.05	93.26	93.94	85.21	88.48	89.47
	DeIT_base	88.88	89.38	91.22	78.77	81.70	85.28
	Swin_base	85.42	87.56	89.78	75.73	79.88	82.46
	ResNet50	70.49	71.75	69.61	57.74	58.48	59.60
	VGG16	88.75	89.34	91.76	79.04	82.53	85.63
Reptile	MViT	80.60	80.20	84.30	72.00	73.20	78.87
	ViT_tiny	85.60	88.00	90.10	75.27	82.47	86.00
	ViT_small	86.80	90.70	93.90	77.73	82.27	87.00
	ViT_base	77.10	81.50	88.00	70.53	78.27	88.07
	DeIT_base	84.40	87.30	92.50	76.20	82.33	86.27
	Swin_base	81.40	87.20	87.40	80.47	81.53	87.87
	ResNet50	80.90	82.00	89.30	73.67	75.87	81.73
	VGG16	84.90	88.80	93.20	77.73	81.67	88.60
MatchingNet	MViT	80.10	81.97	85.32	67.61	72.27	76.35
	ViT_tiny	86.00	89.09	90.91	77.34	80.92	83.93
	ViT_small	90.84	92.56	94.27	84.74	87.02	89.23
	ViT_base	89.56	89.50	92.10	78.24	82.53	86.33
	DeIT_base	89.25	89.36	91.70	79.43	82.39	85.16
	Swin_base	82.01	83.58	86.94	70.34	74.38	78.27
	ResNet50	76.01	74.46	77.89	59.66	61.23	64.08
	VGG16	87.66	88.30	89.94	77.07	79.60	83.45

datasets and few-shot learning tasks, and reduced algorithmic complexity, ProtoNet or MatchingNet with a ViT backbone appears to be a more favorable choice than a CNN or a ViT combined with Reptile. As ViT_small with ProtoNet showed the highest performance across all datasets, it will be our primary model in the following sections. ResNet50 will be its CNN counterpart with a similar size despite demonstrating a lower performance.

4.4.3 Augmentation Results

This study examined the Cutout, Mixup, and Cutmix augmentation techniques on the ISIC 2018 dataset. The findings are compiled in Table 10. For the ProtoNet method, Cutout was the only applicable technique, which, unfortunately, led to reduced scores for most tasks when implemented with both ViT_small and ResNet50. When it comes

Table 10. Effect of different Augmentation techniques on Few-shot classification for ISIC 2018 Dataset

Algorithm	Model	FSL	2-way			3-way		
			3 shot	5 shot	10 shot	3 shot	5 shot	10 shot
ProtoNet	ViT_small	Standart	**84.35**	**86.70**	**89.72**	**72.10**	76.18	**81.45**
		CutOut	81.73	85.89	89.22	70.55	**76.23**	81.13
		MixUp	-	-	-	-	-	-
		CutMix	-	-	-	-	-	-
	ResNet50	Standart	**66.62**	68.65	**72.81**	**51.43**	**53.83**	**58.34**
		CutOut	65.52	**68.75**	72.18	49.32	53.81	57.74
		MixUp	-	-	-	-	-	-
		CutMix	-	-	-	-	-	-
Reptile	ViT_small	Standart	76.05	80.30	85.55	**67.50**	**73.15**	77.37
		CutOut	75.30	**80.35**	83.95	64.87	69.97	76.53
		MixUp	**77.50**	79.40	**85.75**	66.20	71.33	77.87
		CutMix	74.85	77.75	85.65	67.40	72.57	**79.63**
	ResNet50	Standart	70.28	**75.78**	**78.83**	54.47	58.22	61.58
		CutOut	68.73	73.60	76.58	**55.70**	59.90	64.67
		MixUp	**70.75**	74.15	78.03	55.00	**60.65**	**64.95**
		CutMix	70.10	74.60	77.95	53.70	58.92	63.62

Table 11. Cross-domain with MiniIN as a source and medical datasets as targets

Target	Algorithm	2-way			3-way		
		3-shot	5-shot	10-shot	3-shot	5-shot	10-shot
ISIC2018	CD + PN	76.67	79.73	84.62	62.84	67.49	72.93
	Non CD	**84.35**	**86.70**	**89.72**	**72.10**	**76.18**	**81.45**
BreakHis X100	CD + PN	74.53	78.02	83.33	64.53	67.71	74.20
	Non CD	**80.64**	**83.80**	**87.62**	**69.39**	**75.91**	**81.47**
Pap Smear	CD + PN	91.90	93.27	94.70	85.93	87.62	89.21
	Non CD	**92.40**	**94.05**	**94.90**	**86.38**	**89.09**	**90.62**

Table 12. Cross-domain ISIC2018-to-BreakHis and -Pap smear

Target	Setting	2-way			3-way		
		3-shot	5-shot	10-shot	3-shot	5-shot	10-shot
BreakHis X100	CD + PN	74.12	78.74	84.11	62.72	68.90	74.72
	Non CD	**80.64**	**83.80**	**87.62**	**69.39**	**75.91**	**81.47**
Pap Smear	CD + PN	**92.22**	**94.12**	94.85	86.22	88.82	90.47
	Non CD	92.40	94.05	**94.90**	**86.38**	**89.09**	**90.62**

to Reptile, the outcomes were slightly more favorable. The employment of Cutout resulted in a decline in performance for most tasks, except for ResNet50's 3-way k-shot tasks. CutMix demonstrated similar trends, with lower results for most tasks. In contrast, the Mixup technique enhanced the accuracy scores in 4 out of 6 tasks for ResNet50 and 3 tasks for ViT_small when used for data augmentation. Overall, Mixup outperformed the other techniques, making it a commendable data augmentation technique.

4.4.4 Cross-domain Results

This section delves into the results of two categories of cross-domain experiments conducted in this study: natural-medical and medical-medical. In the first category, MiniImageNet was utilized as the source dataset, while ISIC2018, Pap smear, and BreakHis X100 datasets served as target datasets. In the second category, ISIC2018 was the source dataset, and the remaining medical datasets (Pap smear and BreakHis X100) were the target datasets. Results are presented in Tables 11-12

A comparative analysis of the results from both types of cross-domain experiments reveals a drop in performance compared to the non-cross-domain case, suggesting that

Table 13. Comparison with MetaMed and PFEMed on ISIC 2018

Algorithm	Model	2-way			3-way		
		3-shot	5-shot	10-shot	3-shot	5-shot	10-shot
ProtoNet	ViT_small	**84.35**	**86.70**	**89.72**	**72.10**	**76.18**	**81.45**
	ResNet50	66.62	68.65	72.81	51.43	53.83	58.34
MatchingNet	ViT_small	78.40	81.61	86.34	65.50	70.00	76.47
	ResNet50	67.99	71.69	75.66	52.93	56.53	61.60
Reptile	ViT_small	76.05	80.30	85.55	67.50	73.15	77.37
	ResNet50	70.28	75.78	78.83	54.47	58.22	61.58
	MetaMed[20]	72.75	75.62	81.37	54.83	59.33	69.75
-	PFEMed[6]	81.69	83.87	85.14	66.94	69.78	73.81

Table 14. Comparison with MetaMed and PFEMed on BreakHis

Algorithm	Model	2-way			3-way		
		3-shot	5-shot	10-shot	3-shot	5-shot	10-shot
ProtoNet	ViT_small	80.64	83.80	87.62	**69.39**	**75.91**	81.47
	ResNet50	68.62	72.12	73.31	55.80	60.28	61.88
MatchingNet	ViT_small	76.53	82.09	**88.33**	67.13	72.88	**81.80**
	ResNet50	73.45	76.58	79.14	59.53	62.70	66.98
Reptile	ViT_small	68.10	75.60	81.60	54.40	63.13	72.20
	MetaMed[20]	78.75	81.38	83.88	63.08	66.42	74.08
-	PFEMed[6]	**82.16**	**85.28**	86.90	69.21	75.04	78.93

Table 15. Comparison with MetaMed and PFEMed on Pap smear

Algorithm	Model	2-way			3-way		
		3-shot	5-shot	10-shot	3-shot	5-shot	10-shot
ProtoNet	ViT_small	92.40	94.05	94.90	86.38	89.09	90.62
	ResNet50	70.49	71.75	69.61	57.74	58.48	59.60
MatchingNet	ViT_small	90.84	92.56	94.27	84.74	87.02	89.23
	ResNet50	76.01	74.46	77.89	59.66	61.23	64.08
Reptile	ViT_small	83.35	87.05	91.96	72.52	81.13	87.94
	ResNet50	71.44	74.59	78.39	48.00	49.86	50.44
	MetaMed[20]	85.37	86.50	89.37	70.58	72.42	83.00
-	PFEMed[6]	**95.53**	**95.87**	**96.00**	**92.42**	**92.48**	**92.68**

cross-domain few-shot learning in this context presents challenges. This drop in performance could be attributed to the inherent dissimilarities between the source and target domains, making the transfer of learning more complicated. However, when comparing the two types of cross-domain experiments, it was observed that the medical-medical experiments showed slightly superior performance compared to the natural-medical ones. This could be because medical datasets might share more common characteristics or features, making the cross-domain adaptation between them slightly more feasible than when adapting from a natural image dataset to a medical one.

4.4.5 Comparison with State-of-The-Art

In this section, we compare the performance of our models with those reported in the MetaMed [20] and PFEMed [6]. We focused on the ViT_small and ResNet50 models, both of which were used with ProtoNet, MatchingNet, and Reptile. The outcomes are articulated in Tables 13 through 15. It is crucial to highlight that all models were meta-trained, devoid of augmentation techniques. However, it is also noteworthy that MetaMed employed a simple CNN model (with only 3840 parameters), a standard in few-shot learning, while PFEMed implemented a model with 72.95M parameters, significantly larger than the 22M and 23.5M parameters of ViT_small and ResNet50 respectively.

On examining the results across all datasets, it was observed that ViT_small surpassed the other models in all tasks when paired with ProtoNet on the ISIC 2018 dataset. On the BreakHis X100 dataset, it achieved the highest accuracy in the 2-way-10 shot and all 3-way tasks. However, on the Pap smear dataset, PFEMed outperformed all other models across all tasks. In general, it appears that the ViT outcomes scale more favorably with an increase in the number of shots compared to PFEMed. In contrast, ResNet50 lagged behind other models' performance.

5. Conclusion

In this paper, we investigated the use of the ViT model in medical image classification, specifically within an FSL framework. We examined various ViT and CNN models, employing the ProtoNet and Reptile algorithms on three benchmark medical datasets: ISIC 2018, BreakHis, and Pap smear. Our findings demonstrated that utilizing ViT as the backbone for ProtoNet outperformed other setups, including configurations that involved ResNet50. We also benchmarked our results against other prominent studies in the field. The pairing of ViT_small with ProtoNet surpassed the outcomes presented in other works. These observations suggest that when combined with ProtoNets, ViTs can be a highly effective tool for few-shot medical image classification tasks. Additionally, we examined the impact of augmentation techniques on the performance of the ViT_small and ResNet50 models using the ISIC 2018 dataset. Only Mixup positively affected model performance among other techniques, improving test scores in 4 and 3 tasks out of 6 for ResNet50 and ViT_small models, respectively, when used with the Reptile algorithm. Meanwhile, Cutout, the only one compatible with ProtoNet, decreased performance in most cases. In cross-domain experiments, we performed two types of tasks: natural-medical and medical-medical. The results showed that both setups performed less effectively than the non-cross-domain scenario, signifying the challenges in cross-domain FSL. Nevertheless, the medical-medical setup showed slightly better performance than the natural-medical setup, suggesting that domain similarity might play a role in cross-domain learning performance.

Looking toward future research, we intend to explore data generation techniques such as GANs for input augmentation. We also aim to design a specific ViT-based architecture for FSL tasks.

Acknowledgments

This research was funded by Nazarbayev University under Faculty Development Competitive Research Grant Program (No. 11022021FD2925).

References

[1] https://github.com/rwightman/pytorch-image-models. 5

[2] Wei-Yu Chen, Yen-Cheng Liu, Zsolt Kira, Yu-Chiang Frank Wang, and Jia-Bin Huang. A closer look at few-shot classification. *arXiv preprint arXiv:1904.04232*, 2019. 1, 2

[3] Yudong Chen, Chaoyu Guan, Zhikun Wei, Xin Wang, and Wenwu Zhu. Metadelta: A meta-learning system for few-shot image classification. In *AAAI Workshop*, pages 17–28. PMLR, 2021. 6

[4] Yuzhong Chen, Zhenxiang Xiao, Lin Zhao, Lu Zhang, Haixing Dai, David Weizhong Liu, Zihao Wu, Changhe Li, Tuo Zhang, Changying Li, et al. Mask-guided vision transformer (mg-vit) for few-shot learning. *arXiv preprint arXiv:2205.09995*, 2022. 2

[5] Mehdi Cherti and Jenia Jitsev. Effect of pre-training scale on intra-and inter-domain full and few-shot transfer learning for natural and medical x-ray chest images. *arXiv preprint arXiv:2106.00116*, 2021. 2

[6] Zhiyong Dai, Jianjun Yi, Lei Yan, Qingwen Xu, Liang Hu, Qi Zhang, Jiahui Li, and Guoqiang Wang. Pfemed: Few-shot medical image classification using prior guided feature enhancement. *Pattern Recognition*, 134:109108, 2023. 2, 8

[7] Terrance DeVries and Graham W Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017. 2, 4

[8] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 1, 3, 5

[9] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML*, pages 1126–1135. PMLR, 2017. 4

[10] Kelei He, Chen Gan, Zhuoyuan Li, Islem Rekik, Zihao Yin, Wen Ji, Yang Gao, Qian Wang, Junfeng Zhang, and Dinggang Shen. Transformers in medical image analysis: A review. *Intelligent Medicine*, 2022. 1

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015. 5

[12] Timothy Hospedales, Antreas Antoniou, Paul Micaelli, and Amos Storkey. Meta-learning in neural networks: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 44(9):5149–5169, 2021. 2

[13] Shell Xu Hu, Da Li, Jan Stühmer, Minyoung Kim, and Timothy M Hospedales. Pushing the limits of simple pipelines for few-shot learning: External data and fine-tuning make a difference. In *CVPR*, pages 9068–9077, 2022. 1, 2

[14] Jan Jantzen, Jonas Norup, Georgios Dounias, and Beth Bjerregaard. Pap-smear benchmark data for pattern classification. *Nature inspired Smart Information Systems (NiSIS 2005)*, pages 1–9, 2005. 2, 4

[15] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, pages 10012–10022, 2021. 5

[16] Christos Matsoukas, Johan Fredin Haslum, Magnus Söderberg, and Kevin Smith. Is it time to replace cnns with transformers for medical images? *arXiv preprint arXiv:2108.09038*, 2021. 1

[17] Sachin Mehta and Mohammad Rastegari. Separable self-attention for mobile vision transformers, 2022. 5

[18] Alex Nichol, Joshua Achiam, and John Schulman. On first-order meta-learning algorithms. *arXiv preprint arXiv:1803.02999*, 2018. 2, 4

[19] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition, 2015. 5

[20] Rishav Singh, Vandana Bharti, Vishal Purohit, Abhinav Kumar, Amit Kumar Singh, and Sanjay Kumar Singh. Metamed: Few-shot medical image classification using gradient-based meta-learning. *Pattern Recognition*, 120:108111, 2021. 2, 4, 8

[21] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. *NIPS*, 30, 2017. 2, 3

[22] Yisheng Song, Ting Wang, Subrota K Mondal, and Jyoti Prakash Sahoo. A comprehensive survey of few-shot learning: Evolution, applications, challenges, and opportunities. *arXiv preprint arXiv:2205.06743*, 2022. 2

[23] Fabio A Spanhol, Luiz S Oliveira, Caroline Petitjean, and Laurent Heutte. A dataset for breast cancer histopathological image classification. *Ieee transactions on biomedical engineering*, 63(7):1455–1462, 2015. 2, 4

[24] Yonglong Tian, Yue Wang, Dilip Krishnan, Joshua B Tenenbaum, and Phillip Isola. Rethinking few-shot image classification: a good embedding is all you need? In *ECCV*, pages 266–282. Springer, 2020. 1

[25] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *ICML*, pages 10347–10357. PMLR, 2021. 5

[26] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *NIPS*, 30, 2017. 1

[27] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. *NIPS*, 29, 2016. 2, 4

[28] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *ICCV*, pages 6023–6032, 2019. 2, 4

[29] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017. 2, 4

[30] Jinyi Zou, Xiao Ma, Cheng Zhong, and Yao Zhang. Dermoscopic image analysis for isic challenge 2018. *arXiv preprint arXiv:1807.08948*, 2018. 2, 4