META-learning-based retinal pathology classification from optical coherence tomography images

Ziting Yin[1], Xinjian Chen[1,2], Weifang Zhu[1], Dehui Xiang[1], Qing Peng[3], Fei Shi[1,*]
[1]School of Electronics and Information Engineering, Soochow University, Suzhou, 215006, China
[2]State Key Laboratory of Radiation Medicine and Protection, Soochow University, Suzhou, 215123, China
[3]Shanghai Tenth People's Hospital, Shanghai, 200072, China

ABSTRACT

Meta-learning has been proposed with the goal of achieving general artificial intelligence, which makes deep learning models imitate advanced organisms, using prior knowledge to quickly adapt to new learning tasks with just a small number of samples. This ability is especially important for medical image analysis when training samples with pathologies are sometimes limited. To make full use of available medical image data and improve classification results, we propose to apply model-agnostic meta-learning (MAML) and MAML++ for pathology classification from optical coherence tomography (OCT) images. MAML trains a set of initialization parameters using training tasks, by which the model achieves fast convergence in new tasks with only a small amount of data. MAML++ is an improved version, which overcomes some shortcomings of MAML. Our model is pretrained on an OCT dataset with seven types of retinal pathologies, and then refined and tested on another dataset with three types of pathologies. According to the experimental results, the classification accuracies of MAML and MAML++ reached 90.60% and 95.60% respectively, which are higher than the traditional deep learning methods with pretraining.

Keywords: meta-learning, OCT, classification, MAML, MAML++

1. INTRODUCTION

Optical coherence tomography (OCT) is a diagnostic imaging technique that has a wide range of applications in the field of ophthalmology. OCT images can be used by ophthalmologists to make a range of diagnoses, including vitreomacular interface diseases, retinal hemorrhage, macular edema, and chorioretinal diseases, among other ophthalmic conditions. Existing deep learning methods have achieved great success in the classification of OCT images[1,2]. However there are clear limitations. For example, successes have largely been in areas where vast quantities of data can be collected or simulated, and where huge computing resources are available. This excludes many applications where data is intrinsically rare or expensive, or computing resources are unavailable. As in clinical settings, the number of training samples is often limited, especially for rare diseases, and traditional deep learning cannot classify them effectively due to the lack of sufficient training data.

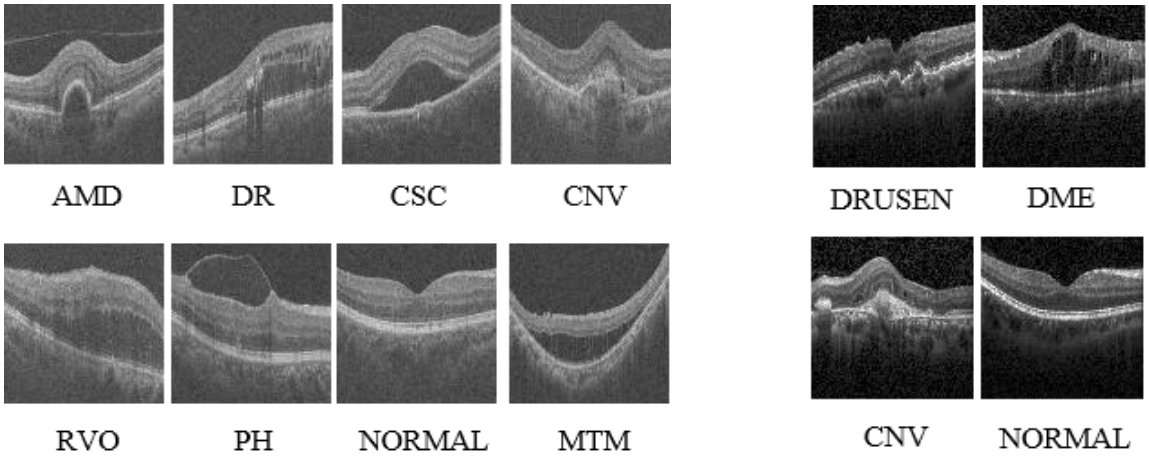
To solve the above problem, we choose to use meta-learning which is attracting rising interest in the field of deep learning. It was proposed to solve the problem of few-shot learning[3] and to obtain a model that can be quickly applied to new tasks with small number of training samples. Meta-learning provides an alternative paradigm where a machine

* email: shifei@suda.edu.cn

learning model gains experience over multiple learning episodes - often covering a distribution of related tasks - and uses this experience to improve its future learning performance[4,5]. This "learning-to-learn"[6] idea can effectively improve the lack of generalization performance of traditional neural network models and their poor adaptability to new kinds of tasks.

A gradient-based meta-learning approach called model-agnostic meta-learning (MAML) was proposed in [7] which is one of the most popular optimization-based meta-learning frameworks, due to its simplicity and good performance in many meta-learning tasks. In addition, MAML++[8], as an improved algorithm of MAML, is proved to achieve better performance in terms of stability, convergence speed, expressiveness and so on.

In this paper, we study the performance of these two meta-learning methods on retinal pathology classification based on OCT images. A convolutional neural network is initialized with an OCT dataset containing seven types of pathologies and normal controls, and then fine-tuned and tested on new tasks from another dataset, which contains three types of pathologies and normal controls. Performance of meta-learning methods are compared with the traditional pre-training method in deep learning. Example images from the two datasets are shown in Figure 1.



(a) (b)

Figure 1. Example B-scans from the two datasets.

2. METHODS

2.1 Meta-learning: learning to initialize

MAML[7], one of the most classical algorithms of meta-learning, trains a set of initialization parameters with groups of training tasks, and obtains a model that achieves fast convergence and good performance with only a small amount of training data from new tasks. MAML can be shown to retain the generality of black-box meta-learners such as RNNs [9], while being applicable to standard neural network architectures.

Specifically, MAML aims to find a learning algorithm called F, and trains on a batch of tasks by this F to generate the initialization parameters θ. When encountering a new task T_i, θ is updated to θ_i correspondingly, forming an exclusive parameter for the new task T_i. In a word, if there is a suitable initialization parameter θ, it can be optimized only one or a few steps at a time to turn into θ_i quickly for a new task with only a few number of training samples, and well implemented on the new task.

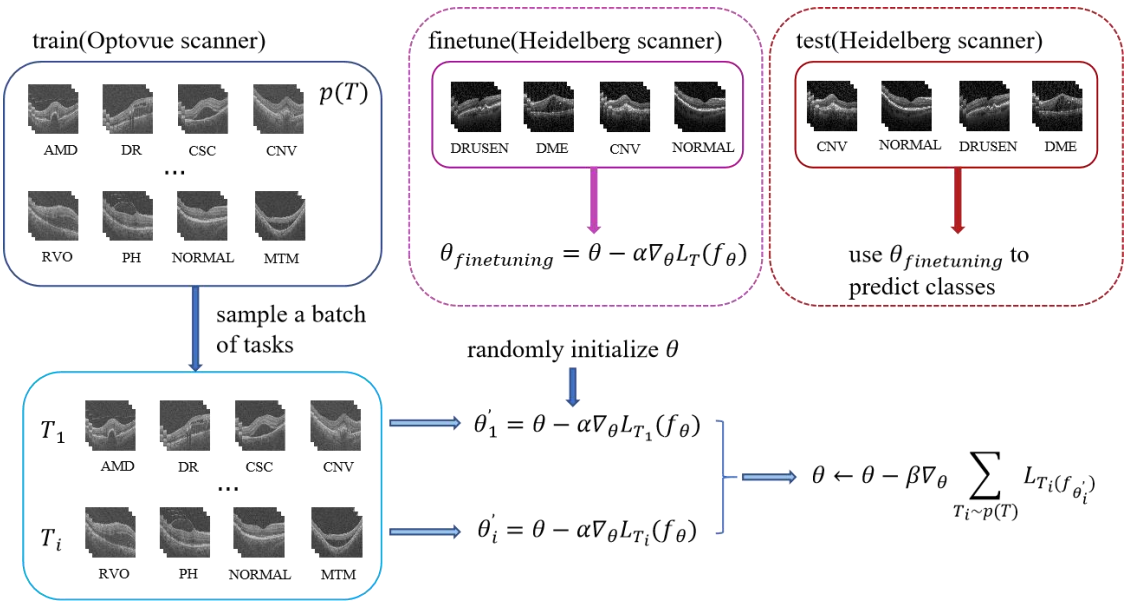


Figure 2. Training Process of MAML

Different from conventional machine learning, the training sample in meta-learning is task rather than data instance, which means slicing and dicing an existing dataset into multiple tasks. Usually it works in a few-shot learning setting[3], where a task is defined as an N-way K-shot problem consisting of a support set and a query set. Both the support set and the query set include the same N categories, where the support set has K images and the query set has Q images for each category, respectively.

2.2 MAML

The MAML algorithm consists of two loops: an inner loop and an outer loop. The optimal parameters for each task are found in the inner loop. The outer loop updates the randomly initialized model parameters by computing the gradient relative to the optimal parameters in each new task. The whole training process of MAML is shown in Figure 2.

First, given a distribution over tasks $p(T)$, meta tasks $T_i\,(i \in 1, \ldots M)$ are randomly selected to build a batch of tasks. Each task contains N categories, with K images in each category as support sets. For each task in a batch, the parameters θ are updated to $\theta_i^{'}$ by one or several steps of gradient descent as in Eq(1). We set up 5 steps in our experiments.

$$\theta_i^{'} = \theta - \alpha\nabla_\theta L_{T_i} f(\theta_i) \tag{1}$$

where α denotes the learning rate, $L_{T_i} f(\theta_i)$ denotes the loss on support set of T_i. In the outer loop, the model parameter θ is updated by calculating gradients with respect to the sum of all losses of $\theta_i^{'}$ obtained in the inner loop, as in Eq(2),

$$\theta \leftarrow \theta - \beta\nabla_\theta \textstyle\sum_{T_i \sim p(T)} L_{T_i}(f_{\theta_i^{'}}) \tag{2}$$

where β denotes the learning rate, and $L_{T_i}(f_{\theta_i^{'}})$ denotes the loss on query set of T_i.

In the MAML algorithm, the cross-entropy is used as the loss function. The loss takes the form as follows:

$$L_{T_i}(f_\phi) = \textstyle\sum_{\boldsymbol{x}^{(j)}, \boldsymbol{y}^{(i)} \sim T_i} \boldsymbol{y}^{(i)} \log f_\phi(\boldsymbol{x}^{(i)}) + (1 - \boldsymbol{y}^{(i)}) \log (1 - f_\phi(\boldsymbol{x}^{(i)})) \tag{3}$$

where $\boldsymbol{x}^{(i)}$ denotes the input images from task T_i, $f_\phi(\boldsymbol{x}^{(i)})$ and $\boldsymbol{y}^{(i)}$ denotes the predicted and true labels of the images.

2.3 MAML++

MAML is one of the best methods available for meta-learning with few samples. MAML is generally simple and powerful. However, it has various problems such as very sensitive neural network structure which often leads to unstable training, the need for painstaking hyperparameter search to achieve stable training and good generalization and the need for very expensive training and inference time. Therefore, in MAML++[8], various modifications have been proposed to MAML that not only stabilize the system but also greatly improve the generalization performance, convergence speed and computational overhead of MAML.

Specifically, MAML++ points out six problems of MAML as follows, and then proposes corresponding solutions for improvement.

Training Instability: MAML minimizes the set loss computed by the base-network after completing the internal loop update of the support set, leading to many instability problems such as gradient diminishing and gradient explosion. Therefore MAML++ proposes the multi-step loss, which minimizes the target set loss at each step of the inner loop as in Eq(4). In addition, the annealing strategy is used to weight the loss per step.

$$\theta = \theta - \beta \nabla_{\theta} \sum_{b=1}^{B} \sum_{i=0}^{N} v_b L_{T_i}(f_{\theta_i^b}) \tag{4}$$

where $L_{T_i}(f_{\theta_i^b})$ denotes the target set loss of task T_i when using the base-network weights after b steps towards minimizing the support set task and v_b denotes the importance weight of the target set loss at step b.

Second Order Derivative Cost: MAML uses first-order derivative approximation for the whole training in order to improve the computational efficiency, which is one of the main reasons affecting the generalization of the model. The authors of MAML++ found through experiments that using first-order derivatives for the first fifty epochs and then switching to using second-order derivatives could get better results. Moreover, there is no gradient explosion or gradient vanishing, which is more stable than just using second-order derivatives.

Absence of Batch Normalization Statistic Accumulation: MAML++ uses per-step batch normalization running statistics (BNRS) to make improvements to MAML's lack of batch normalization statistic accumulation, which can accelerate MAML optimization and potentially improve generalization performance.

Shared Batch Normalization Bias: MAML uses the same bias in all iterations of the base model. This implicitly assumes that all base models are the same throughout the inner-loop update process, and therefore the distribution of features passing through them is the same. But this is actually wrong. MAML++ improves it by learning a set of bias at each step of the inner-loop update process. In this way, batch normalization will learn deviations specific to the distribution of features seen at each set, which will improve convergence speed, stability, and generalization performance.

Shared Inner Loop Learning Rate: MAML++ proposes to learn the learning rate and direction for each layer in the network, as well as learning a different learning rate as the underlying network gradually adapts. Learning the learning rate and direction per layer rather than per parameter reduces the memory and computation required, while providing more flexibility in the update step.

Fixed Outer Loop Learning Rate: In MAML the Adam optimizer with a fixed learning rate is used to optimize the meta-objective. The fixed learning rate makes the adjustment of hyper-parameters very difficult and inflexible. To fix the problem, MAML++ uses the Cosine Annealing (CA) algorithm to make the outer loop learning rate dynamically variable and to improve the performance of the algorithm.

3. EXPERIMENTS AND RESULTS

3.1 Datasets

Dataset 1 and 2 used in this paper are two-dimensional OCT B-scans extracted from volumetric scans, which are from the publicly available OCTA-500[10,11] and Large Dataset of Labeled Optical Coherence Tomography (OCT) and Chest X-Ray Images[1,12], respectively. The OCTA-500 dataset contains 3D data in both OCT and OCTA modalities acquired by the Optovue scanner, from which we only use the OCT modality. The pathology classes include age-related macular degeneration (AMD), diabetic retinopathy (DR), choroidal neovascularization (CNV), central serous chorioretinopathy (CSC) and retinal vein occlusion (RVO). In addition, there are NORMAL samples and a few samples with myopic traction maculopathy (MTM) and preretinal hemorrhage (PH) under the OTHERS category. We use the above eight categories of OCT images in OCTA-500. Large Dataset of Labeled Optical Coherence Tomography (OCT) and Chest X-Ray Images contains OCT B-scans acquired by the Heidelberg Spectralis scanner, belonging to four categories: CNV, diabetic macular edema (DME), DRUSEN, and NORMAL, and the images are split into independent training and testing sets.

In our experiment, we use Dataset 1 to pretrain the model and then refine and test it on Dataset 2. Specifically, Dataset 1 contains 80 samples from each class as training data for meta-learning, constituting 4-way 5-shot 3-query tasks, and 10 tasks are taken as a batch for training. Dataset 2 contains independent training and testing sets. 64 samples from each class are used for fine-tuning, which also constitutes 4-way 5-shot 3-query tasks with a batchsize of 8. Since the number of categories in the test set is four, the tasks in both training and fine-tuning are built with four categories, which are consistent with the test data. We select five images for each category as the support set and three images as the query set, which puts each task in the few-shot learning setting. For final test, 50 samples from each class are used.

For comparison, we apply a traditional deep learning method which consists a pre-training stage on Dataset 1 and a fine-tuning stage on Dataset 2. We construct the deep learning dataset, where 50 and 30 samples from each class in Dataset 1 are used for training and validation in the pre-training stage, respectively. Similarly, 40 and 24 samples from each class in Dataset 2, are used for training and validation in the fine-tuning stage. Fine-tuning here means fine-tuning the initialization parameters obtained from the pre-training process with a new dataset, and taking out the model with the best results in the fine-tuning for testing.

Table 1. Datasets for Meta-learning and Deep-learning

<table>
<tr><td rowspan="2">datasets</td><td colspan="2">Dataset 1
8×80 B-scans</td><td colspan="2">Dataset 2
4×64 B-scans</td><td>Dataset 2
4×50 B-scans</td></tr>
<tr><td rowspan="3">Meta-learning
(MAML
/MAML++)</td><td colspan="2">train(images/category)</td><td colspan="2">fine-tuning(images/category)</td><td rowspan="2">test
(images/category)</td></tr>
<tr><td>support set</td><td>query set</td><td>support set</td><td>query set</td></tr>
<tr><td>5shot
batch=10
total=50</td><td>3query
batch=10
total=30</td><td>5shot
batch=8
total=40</td><td>3query
batch=8
total=24</td><td>50</td></tr>
<tr><td rowspan="3">Deep-learning</td><td colspan="2">pretrain</td><td colspan="2">fine-tuning</td><td rowspan="2">test
(images/category)</td></tr>
<tr><td>train(images/
category)</td><td>val(images/
category)</td><td>train(images
/category)</td><td>val(images/
category)</td></tr>
<tr><td>50</td><td>30</td><td>40</td><td>24</td><td>50</td></tr>
</table>

In meta-learning and deep learning, the number of images used in the pre-training stage matches that of meta training, and the number of images used in fine-tuning also matches, while they are randomly divided. The final test set is exactly the same as in the meta-learning method. The details of data arrangements are shown in Table 1.

3.2 Implementation details

The model used in experiment is the base network in [7], which has 4 modules with a 3×3 convolutions and 64 filters, followed by batch normalization, a ReLU non-linearity and 2×2 max-pooling. Finally a linear layer and softmax operation are used to give class predictions. Cross-entropy is used as the loss function for all experiments.

In MAML, the models were trained using the Adam optimizer with the fixed learning rates of α=0.003 and β=0.0002. Similarly, the models in MAML++ were also trained using the Adam optimizer with the initial learning rates of α= 0.001 and β=0.0002. These learning rates in MAML++ were automatically adjusted during the training process. With deep learning, the models was trained using the Adam optimizer with a learning rate of 0.0005. The models of MAML and MAML++ were trained both by 250 epochs, while the model of deep learning was trained by 100 epochs. The implementation of the proposed framework was based on the public platform PyTorch and on a NVIDIA GeForce RTX 2080Ti graphics card with 11GB of video memory. As the split of data in pretraining and fine-tuning for both meta-learning and deep-learning were random, the training were repeated for 5 times for all methods, and the mean accuracy with standard deviation is calculated.

3.3 Experimental results

Table 2 shows the classification results of different methods. It is clearly that meta-learning has a higher accuracy rate compared to traditional deep-learning. The accuracy of deep learning is only 85.00%, while the accuracy of MAML is 90.60%. It shows that meta-learning is more advantageous than deep learning when the samples for fine-tuning are limited. Comparing MAML and MAML++, MAML++ has an accuracy rate of 95.60%, which is 5% more than MAML. This indicates that the modifications of MAML++ is fruitful. Figure 3 shows the confusion matrix of different methods. As can be seen, it is a bit difficult for the deep learning model to classify CNV and DRUSEN. While MAML++ is more accurate in classifying each type of diseases. This also illustrates the superiority of MAML++.

The running time of deep learning and meta-learning is also shown in Table 2. The time here refers to the time required to train one epoch. In terms of running time, the training process of meta-learning is longer compared to the pre-training process of deep learning because meta-learning itself is more computationally intensive. The fine-tuning time of meta-learning is significantly reduced compared to deep learning because deep learning still requires complete training in this process, while meta-learning only requires fine-tuning with the new dataset in this stage. In addition, MAML++ runs slightly longer than MAML in both the training and fine-tuning phases, which is consistent with the theory, because the computation and operation steps of MAML++ are more complex than those of MAML. Overall, MAML++ has a longer training time, but such a sacrifice is worthwhile compared to the accuracy improvement.

Table 2. Classification Results of Different Methods

Methods	Accuracy(%)	Training time(s)	Finetuning time(s)
Deep-learning	85.00±1.17	**4.99**	6.30
MAML	90.60±2.17	11.83	**0.32**
MAML++	**95.60±3.78**	12.17	0.39

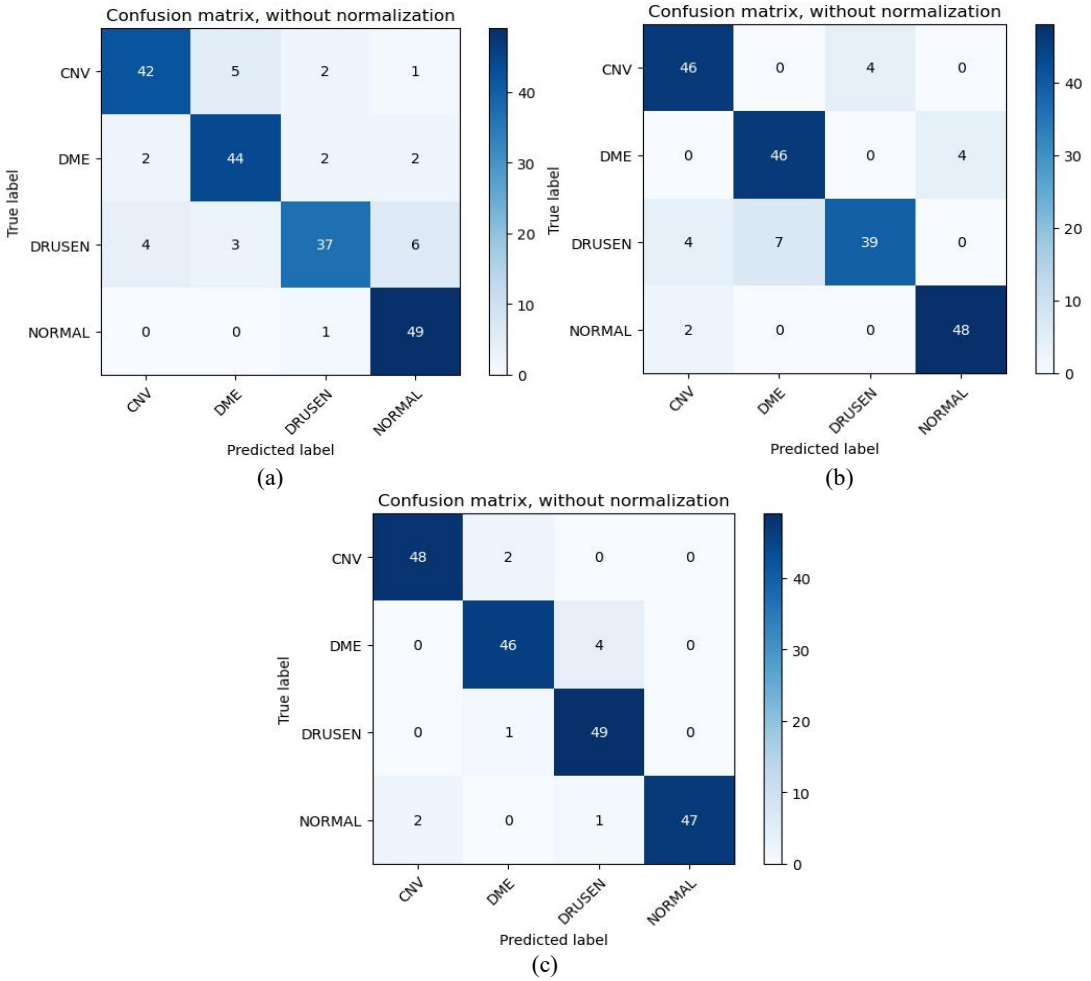


Figure 3. Confusion Matrix of Different Methods. (a) Deep-learning. (b) MAML. (c) MAML++

4. CONCLUSIONS

In this paper, we apply meta-learning for the classification of retinal pathologies in OCT images with small-scale training data. To compare the classification performance of meta-learning, we use the classical algorithm MAML in meta-learning and its improved version MAML++ to compare with deep learning with pre-training. The experiment results show that the classification accuracy of meta-learning is significantly higher than that of traditional deep learning in the case of limited samples, and the improved MAML++ shows even superior performance. This proves that with meta-learning, the model can effectively learn from prior knowledge and perform well on the new task.

In the future, we will explore more improvements on the meta-learning algorithms and on the network architectures, and test meta-learning models on real-case rare retinal diseases.

ACKNOWLEDGEMENTS

This work was supported by the National Key Research and Development Program of China (2018YFA0701700), and the National Natural Science Foundation of China (62271337，61971298).

REFERENCE

[1]D. Kermany, M. Goldbaum M, C. Cai, et al, "Identifying medical diagnoses and treatable diseases by image-based deep learning," Cell, 2018, 172(5): 1122-31 e9.

[2]J. Fauw, J. Ledsam, B. Romera-Paredes, et al, "Clinically applicable deep learning for diagnosis and referral in retinal disease," Nature Medicine, 2018, 24(9): 1342-1350.

[3]Y. Wang, Q. Yao, J. Kwok, et al, "Generalizing from a few examples: a survey on few-shot learning," ACM Computing Surveys,2020, 53(3): 1-34.

[4]J. Vanschoren, "Meta-learning: a survey," arXiv:1810.03548 [cs.LG].

[5]T. Hospedales, A. Antoniou, P. Micaelli, et al, "Meta-learning in neural networks: a survey,", IEEE Transactions on Pattern Analysis and Machine Intelligence, 2022, 44(9): 5149-5169.

[6]M. Huisman, J. Rijn, A. Plaat, "A survey of deep meta-learning," arXiv:2010.03522 [cs.LG].

[7]C.Finn, P. Abbeel, S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," arXiv.1703.03400 [cs.LG].

[8]A. Antoniou, H. Edwards, A. Storkey, "How to train your MAML," arXiv:1810.09502v3 [cs.LG].

[9]C. Finn, S. Levine, "Meta-learning and universality: deep representations and gradient descent can approximate any learning algorithm". arXiv:1710.11622 [cs.LG].

[10]M. Li, Y. Chen ,Z. Ji , et al, "Image projection network: 3D to 2D image segmentation in OCTA images," IEEE Transactions on Medical Imaging, 2020, 39(11): 3343-3354.

[11]M. Li, Y. Chen, K. Xie, et al, December 23, 2019, "OCTA-500", IEEE Dataport, https://ieee-dataport.org/open-access/octa-500.

[12]D. Kermany, K. Zhang, M. Goldbaum, June 2, 2018 , "Large dataset of labeled optical coherence tomography (OCT) and chest X-Ray images," Mendeley Data, https://data.mendeley.com/datasets/rscbjbr9sj/3.