Wuhan University of Technology

Complementing Representation Deficiency in Few-shot Image Classification: A Meta-Learning Approach

Xian Zhong
Hubei Key Laboratory of Transportation Internet of Things & School of Computer Science and Technology, Wuhan University of Technology, Wuhan, China
zhongx@whut.edu.cn

Cheng Gu
School of Computer Science and Technology, Wuhan University of Technology, Wuhan, China
gucheng hs@whut.edu.cn

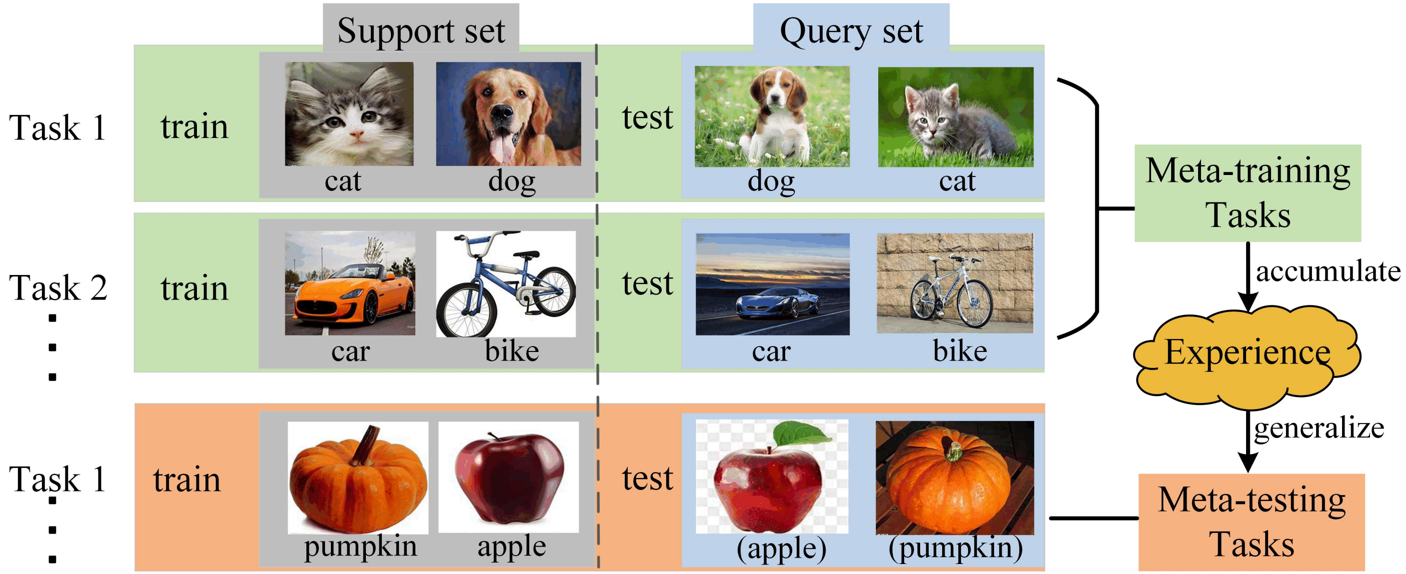
Wenxin Huang
School of Computer Science Wuhan University Wuhan, China
wenxin.huang@whu.edu.cn

Lin Li
School of Computer Science and Technology, Wuhan University of Technology, Wuhan, China
cathylilin@whut.edu.cn

Shuqin Chen
School of Computer Science and Technology, Wuhan University of Technology, Wuhan, China
csqcwx0801@whut.edu.cn

Chia-Wen Lin
Department of Electrical Engineering & Institute of Communications Engineering, National Tsing Hua University, Hsinchu, Taiwan
cwlin@ee.nthu.edu.tw

Few-shot learning



Meta-learning contains a base-learner, and a meta-learner which adapts the base-learner to new tasks with few samples.

Problems

1.It is hard to acquire plenty of samples.

2.**The Representation Deficiency** commonly exists in few-shot learning.

Motivation

- The out-performance of variational inference in generating extra information.

Contribution

- We propose an end-to-end framework and interpolate a latent space to endue the reconstructed latent codes with more information, complementing the representation deficiency in a high-dimensional parameter space.
- The probabilistic latent space with stochastic initialization collaborates well with different base-learners and can be extended to other architecture with high-dimensional feature extractors in few-shot learning.
- We optimize the framework leveraging new loss function for the proposed latent space, which acquires better generalization across tasks and achieves the state-of-theart performance in few-shot learning classification tasks.

Related work

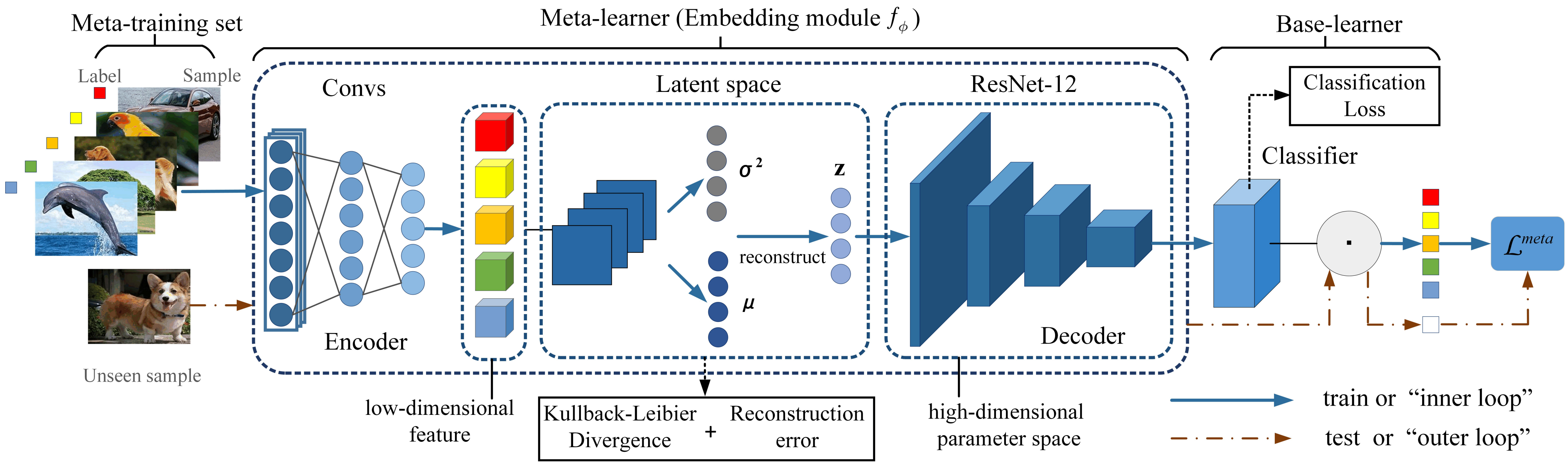
1) **Metric-based methods**, which learn a similarity space through training similar metrics over the same class to enhance effectiveness.

2) **Memory-based methods**, which use memory architecture to store key "experience" from seen samples and generalize to unseen tasks according to the stored knowledge.

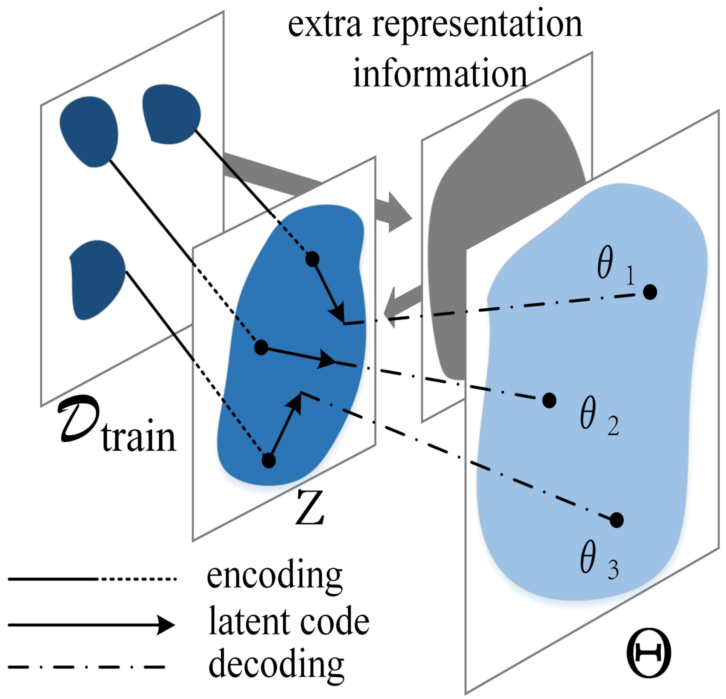
3) **Optimization-based methods,** which search for a suitable meta-learner that is conducive to fast gradient-based adaptation to new tasks.

武漢理工大學
Wuhan University of Technology

Method



Method



The latent code is endued extra information, when reconstructed in latent space Z, to complement the representation deficiency.

Method

Algorithm 1 MCRNet

Input: Meta-training set S^{train}; Encoding function f_{ϕ_e}, and decoding function f_{ϕ_d}; Learning rate λ, η

Initialize ϕ_e, ϕ_d randomly
Let $\phi = \{\phi_e, \phi_d, \lambda\}$
while new batch **do**
  for number of task in batch **do**
    Sample task $\mathcal{T}_i = (\mathcal{D}^{train}, \mathcal{D}^{test})$ from S^{train}
    for number of sample in support set $\mathcal{D}^{train}$ **do**
      Encode samples in $\mathcal{D}^{train}$ to **z** with complemented representations using f_{ϕ_e}
      Decode **z** using f_{ϕ_d}
      Compute θ in base-learner using (6)
      Compute training loss $\mathcal{L}^{train}_{\mathcal{T}_i}(\mathcal{D}^{train}; \phi, \theta, \varphi)$
      Perform gradient optimization:
      $\theta \leftarrow \theta - \lambda \bigtriangledown_\theta \mathcal{L}^{train}_{\mathcal{T}_i}(\mathcal{D}^{train}; \phi, \theta, \varphi)$
    end for
    for number of sample in query set $\mathcal{D}^{test}$ **do**
      Compute test loss $\mathcal{L}^{test}_{\mathcal{T}_i}(\mathcal{D}^{test}; \phi, \theta, \varphi)$
    end for
    Compute meta-training loss $\mathcal{L}^{meta}(\mathcal{S}^{train}; \theta, \phi)$
    Perform gradient optimization:
    $\phi \leftarrow \phi - \eta \bigtriangledown_\phi \mathcal{L}^{meta}(\mathcal{S}^{train}; \theta, \phi)$
  end for
end while

$$\mathcal{L}^{train}_{\mathcal{T}_i}(\mathcal{D}^{train}; \phi, \theta, \varphi) = \sum_{(x_n, y_n) \in D^{train}_i} [-\varphi w_{y_n} \cdot f_\phi(x_n) + \log \sum_{k=1}^{K} \exp(\varphi w_k \cdot f_\phi(x_n))], \tag{8}$$

$$\mathcal{L}^{meta}(\mathcal{S}^{train}; \theta, \phi) = \sum_{\mathcal{T}_i \sim \mathcal{S}^{train}} [\mathcal{L}^{test}_{\mathcal{T}_i}(\mathcal{D}^{test}; \phi, \theta, \varphi) + \beta \mathcal{L}_{var}] \tag{9}$$

Experiments

TABLE I: Comparisons of average classification accuracy (%) with 95% confidence intervals on the CIFAR-FS and FC100. "SVM" or "RR" means using SVM or Ridge Regression as base-learner.

method	backbone	CIFAR-FS		FC100	
		1-shot	5-shot	1-shot	5-shot
Relation Networks [20]	4CONV	55.0 ± 1.0	69.3 ± 0.8	-	-
Prototypical Networks [12]	4CONV	55.5 ± 0.7	72.0 ± 0.6	35.3 ± 0.6	48.6 ± 0.6
MAML [10]	4CONV	58.9 ± 1.9	71.5 ± 1.0	-	-
R2D2 [17]	4CONV	65.3 ± 0.2	79.4 ± 0.1	-	-
Fine-tuning [28]	ResNet-12	64.66 ± 0.73	82.13 ± 0.50	37.52 ± 0.53	55.39 ± 0.57
TADAM [18]	ResNet-12	-	-	40.1 ± 0.4	56.1 ± 0.4
MTL [5]	ResNet-12¶	-	-	**43.6 ± 1.8**	55.4 ± 0.9
Baseline-RR [25]	ResNet-12	72.6 ± 0.7	84.3 ± 0.5	40.5 ± 0.6	55.3 ± 0.6
Baseline-SVM [25]	ResNet-12	72.0 ± 0.7	84.2 ± 0.5	41.1 ± 0.6	55.5 ± 0.6
MCRNet-RR (ours)	ResNet-12	**73.8 ± 0.7**	**85.2 ± 0.5**	40.7 ± 0.6	**56.6 ± 0.6**
MCRNet-SVM (ours)	ResNet-12	**74.7 ± 0.7**	**86.8 ± 0.5**	41.0 ± 0.6	**57.8 ± 0.6**

¶ indicates that the method is not end-to-end.

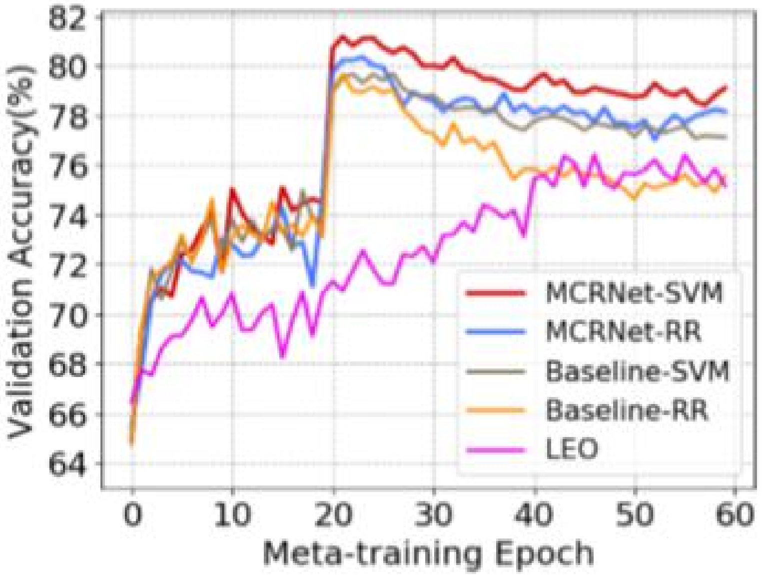
Experiments

TABLE II: Comparisons of average classification accuracy (%) with 95% confidence intervals on the miniImageNet.

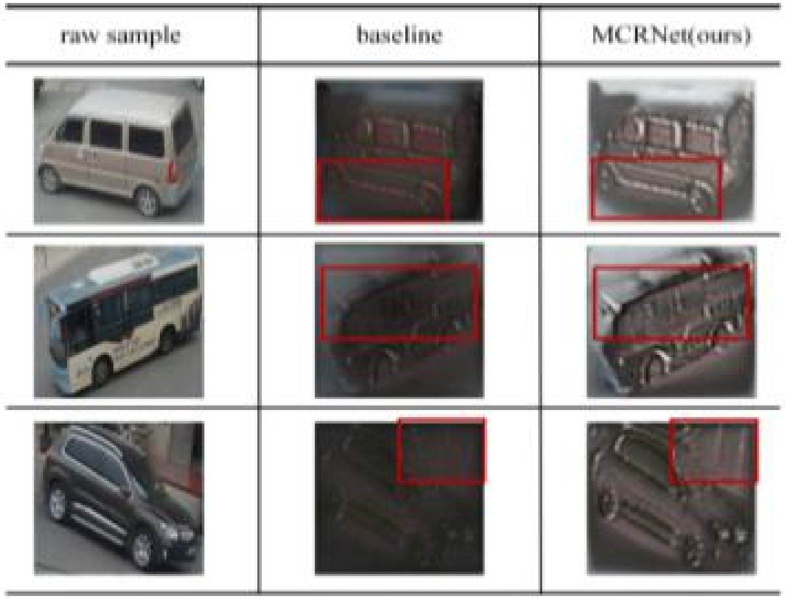
method	backbone	1-shot	5-shot
Meta-Learning LSTM [19]	4CONV	43.44 ± 0.77	60.60 ± 0.71
Matching networks [11]	4CONV	43.56 ± 0.84	55.31 ± 0.73
MAML [10]	4CONV	48.70 ± 1.84	63.11 ± 0.92
Prototypical Networks [12]	4CONV	49.42 ± 0.78	68.20 ± 0.66
Relation Networks [20]	4CONV	50.44 ± 0.82	65.32 ± 0.70
R2D2 [17]	4CONV	51.2 ± 0.6	68.8 ± 0.1
SRAN [15]	ResNet-101¶	51.62 ± 0.31	66.16 ± 0.51
DN4 [29]	ResNet-12	54.37 ± 0.36	74.44 ± 0.29
SNAIL [22]	ResNet-12	55.71 ± 0.99	68.88 ± 0.92
Fine-tuning [28]	ResNet-12	56.67 ± 0.62	74.80 ± 0.51
TADAM [18]	ResNet-12¶	58.50 ± 0.30	76.70 ± 0.30
CAML [30]	ResNet-12	59.23 ± 0.99	72.35 ± 0.71
TPN [31]	ResNet-12	59.46	75.65
wDAE-GNN [32]	WRN-28-10	61.07 ± 0.15	76.75 ± 0.11
MTL [5]	ResNet-12¶	61.2 ± 1.8	75.5 ± 0.8
LEO [3]	WRN-28-10¶	61.76 ± 0.08	77.59 ± 0.12
LEO* [3]	ResNet-12¶	$58.67 \pm 0.07^*$	$73.45 \pm 0.12^*$
Baseline-RR [25]	ResNet-12	$60.02 \pm 0.64^*$	$76.51 \pm 0.49^*$
Baseline-SVM [25]	ResNet-12	$60.73 \pm 0.65^*$	$76.16 \pm 0.49^*$
MCRNet-RR (ours)	ResNet-12	61.32 ± 0.64	$\mathbf{78.16 \pm 0.49}$
MCRNet-SVM (ours)	ResNet-12	$\mathbf{62.53 \pm 0.64}$	$\mathbf{80.34 \pm 0.47}$

¶ indicates those methods are not end-to-end. * indicates those methods that are reproduced by ourselves for comparison of convergence.

Experiments



(a)



(b)

Fig. 4: (a) Comparison of convergence on miniImageNet. Accuracies are obtained on meta-validation set after each epoch. (b) Some examples of the representation output from baseline and MCRNet.

武漢理工大學
Wuhan University of Technology

Experiments

TABLE III: Comparisons of average classification accuracy (%) in different dimensions on the CIFAR-FS, FC100, and miniImageNet.

Latent Dimension	shot	CIFAR-FS		FC100		miniImageNet	
		RR	SVM	RR	SVM	RR	SVM
without MCRNet	1	72.6	72.0	40.5	**41.1**	60.0	60.7
8	1	69.4	70.1	39.0	39.8	57.2	58.7
16	1	70.3	71.2	39.2	40.4	58.4	60.1
32	1	72.9	72.5	**40.7**	40.7	60.3	61.5
64	1	**73.8**	**74.7**	40.4	41.0	**61.3**	**62.5**
without MCRNet	1	84.3	84.2	55.3	55.5	76.5	76.2
8	5	82.6	83.8	53.5	54.0	75.8	75.4
16	5	83.8	84.3	54.0	55.4	76.4	77.5
32	5	84.5	85.5	**56.6**	56.6	76.9	79.0
64	5	**85.2**	**86.8**	55.4	**57.8**	**78.2**	**80.3**

Conclusion

In this paper, we proposed an **MCRNet** for few-shot learning, which achieved state-of-the-art performance on the challenging 5-way 1- and 5-shot CIFAR and ImageNet classification problems.

Future work

1.Decoder: more embedding, such as 4CONV, ResNet-12, WRN-28.

2. Optimization in outer loop.