aMMAI 2021 Final Project

AI 碩二
r08922a20
洪筱慈

Outline

- Motivation
- Used methods
  - Metric learning
  - Multi-source domain
- Experiment results
- Resource & Reference

Motivation

Although we can use a metric loss to make the margin larger, when the unseen domain is distinct from the source domain, the model is still hard to adapt to the target domain

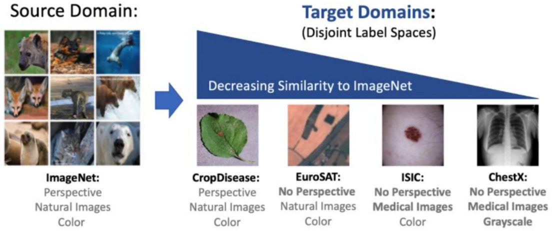


Figure 1: The cross-domain few-shot learning (CD-DSL) benchmark. ImageNet is used for source training, and domains of varying dissimilarity from ImageNet are used for target evaluation. No data is provided for meta-learning, and target classes are disjoint from the source classes.

https://www.learning-with-limited-labels.com/challenge

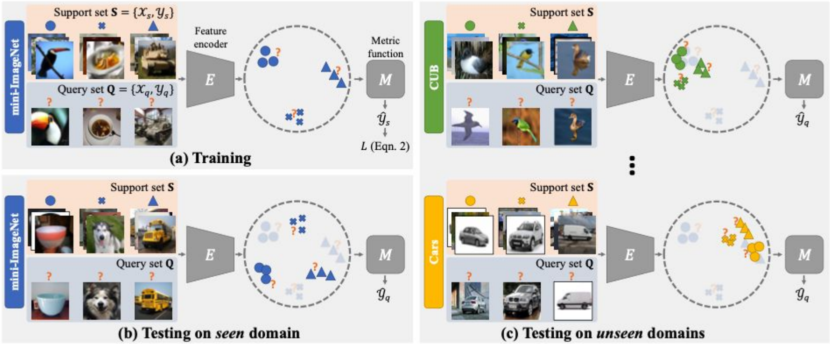
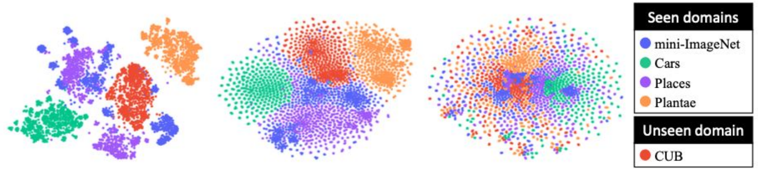


Figure 1: **Problem formulation and motivation.** Metric-based meta-learning models usually consist of a feature encoder E and metric function M. We aim to improve the generalization ability of the models training from seen domains to arbitrary unseen domains. The key observation is that the distributions of the image features extracted from tasks in the unseen domains are significantly different from those in the seen domains.

Tseng et al. Cross-Domain Few-Shot Classification via Learned Feature-Wise Transformation. CVPR 2020

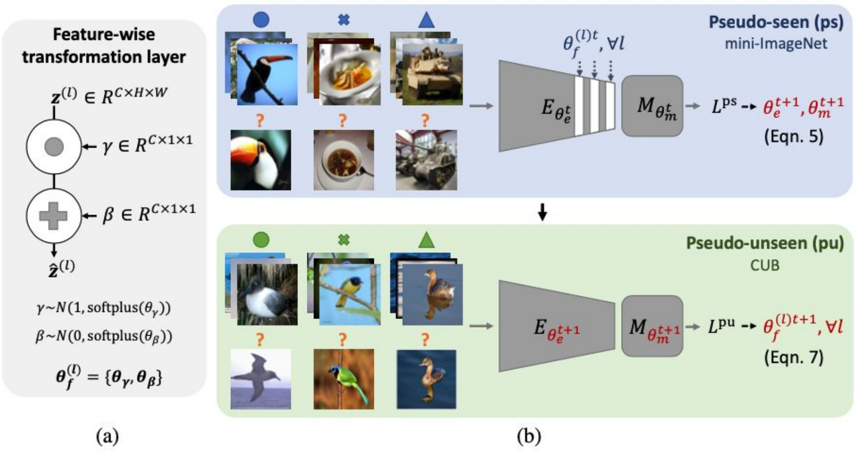
Metric learning

In order to improve the model adaptation ability, use feature-wise transformation layers during pre-training the metric-learning based model



(a) RelationNet (b) RelationNet + FT (c) RelationNet + LFT

Figure 3: **T-SNE visualization of the image features extracted from tasks in different domains.** We show the t-SNE visualization of the features extracted by the (a) original feature encoder E, (b) feature encoder with pre-determined feature-wise transformation layers, and (c) feature encoder with learning-to-learned feature-wise transformation.



(a) (b)

Figure 2: **Method overview.** (a) We propose a feature-wise transformation layer to modulate intermediate feature activation $\mathbf{z}$ in the feature encoder E with the scaling and bias terms sampled from the Gaussian distributions parameterized by the hyper-parameters θ_γ and θ_β. During the training phase, we insert a collection of feature-wise transformation layers into the feature encoder to simulate feature distributions extracted from the tasks in various domains. (b) We design a learning-to-learn algorithm to optimize the hyper-parameters θ_γ and θ_β of feature-wise transformation layers by maximizing the performance of the applied metric-based model on the pseudo-unseen domain (*bottom*) after it is optimized on the pseudo-seen domain (*top*).

Tseng et al, Cross-Domain Few-Shot Classification via Learned Feature-Wise Transformation. CVPR 2020

Metric learning

Yeh et al. proposed to view this problem as an open set problem in face recognition. However, in face recognition, the open set is still belongs to “human face”, which is close to the source domain.
Besides feature transformation layers, how can I improve model generalization ability through loss design?

$$Loss = CrossEntropyLoss + PTLoss$$

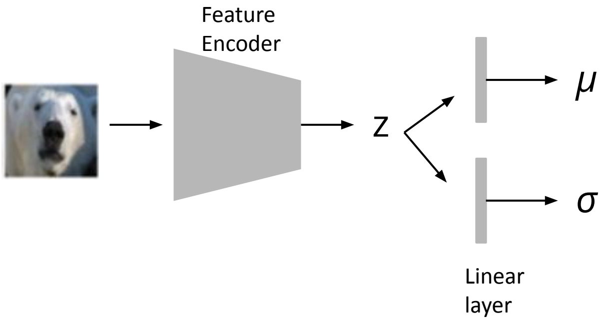
$$PTLoss = \sum_{i=1}^{N \times K} \sum_{j=1}^{N} (c_{s_i} \neq j)\ triplet(s_i,\ p_{c_{s_i}},\ p_j) \quad (1)$$

Yeh et al. Large Margin Mechanism and Pseudo Query Set on Cross-Domain Few-Shot Learning. Cross-Domain Few-Shot Learning CVPR 2020 Challenge

Metric learning with Normal distribution

Inspired by Tseng et al., If the feature space is an normal distribution, then maybe the model can easier to adapt on the target domain

$$Loss = CrossEntropyLoss \; + \; PTLoss \; + \; \beta * KLDLoss$$

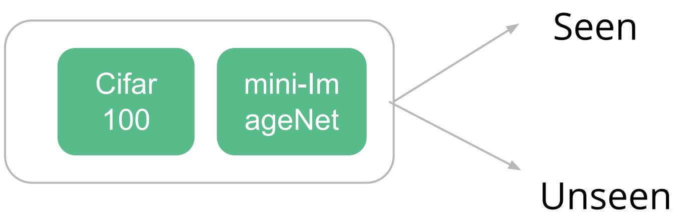


$$PTLoss = \sum_{i=1}^{N \times K} \sum_{j=1}^{N} (c_{s_i} \neq j) \; triplet(s_i, \; p_{c_{s_i}}, \; p_j) \quad (1)$$

$$KLDLoss = \frac{1}{2}\left[-\sum_i (\log \sigma_i^2 + 1) + \sum_i \sigma_i^2 + \sum_i \mu_i^2 \right]$$

Multi-source domain learning

In every epoch during pre-training the metric learning-based model, randomly set one of the two datasets to “pseudo seen” dataset and another to “pseudo unseen” to train the model and the feature transform layers



Experiment results

Backbone: RestNet10

β = 0.1

	fsl	cdfsl-single				cdfsl-multi			
	mini-ImageNet	Crop Disease	EuroSAT	ISIC	Avg.	Crop Disease	EuroSAT	ISIC	Avg.
Baseline	**74.81% ± 0.62%**	88.39% ± 0.54%	78.48% ± 0.63%	49.42% ± 0.58%	72.10%	87.05% ± 0.56%	79.12% ± 0.62%	48.31% ± 0.65%	71.49%
ProtoNet	63.04% ± 0.72%	85.07% ± 0.57%	76.53% ± 0.67%	41.80% ± 0.59%	67.80%	83.84% ± 0.63%	77.41% ± 0.67%	42.94% ± 0.55%	68.06%
PTLoss	65.42% ± 0.65%	87.25% ± 0.50%	76.60% ± 0.68%	43.76% ± 0.57%	69.20%	85.18% ± 0.59%	78.47% ± 0.63%	43.47% ± 0.57%	69.04%
PTLoss + FT	71.65% ± 0.65%	**93.64% ± 0.35%**	84.81% ± 0.53%	**56.23% ± 0.59%**	**78.23%**	**93.17% ± 0.37%**	**86.34% ± 0.52%**	**56.62% ± 0.58%**	**78.71%**
PTLoss + KLD + FT	71.68% ± 0.65%	93.18% ± 0.37%	**85.08% ± 0.50%**	55.65% ± 0.61%	77.97%	92.67% ± 0.38%	85.57% ± 0.50%	55.35% ± 0.57 %	77.86%

- All tracks apart from fsl are improved by PTLoss and PTLoss + FT
- W/ FT outperforms all tracks comparing to w/o FT
- w/o FT, the average accuracy encounters a slight drop from cdfsl-single to cdfsl-multi, as for w/ FT, there is a slight increase (0.48%), which indicated that FT can better leverage the multiple source domain
- w/ FT makes the Standard Error smaller for most of the target domains, indicating that the variability across 600 tasks is smaller
- w/ KLD loss, in cdfsl-single, there is a slight improvement in EuroSAT

Project Resource

- Project info video: https://drive.google.com/file/d/1hcyLAUgfBWM08fTGvz0yASjuPFKaPXEa/view
- Project source code: https://github.com/JiaFong/NTU_aMMAI21_cdfsl
- Project doc: https://docs.google.com/document/d/1VtVR45wBvQlSnap1AArqIvnjoUQeKdWFkWe9EksLEIw/edit
- Project demo list:

  https://docs.google.com/spreadsheets/d/1l9iCLmlpja1U2pcSWrP-hdZvplw-8wpZJl70Rl9C1es/edit#gid=0
- Few-shot lecture video: https://drive.google.com/file/d/1uxj0JOqKOXYlx7FGWoYqcAtSgPpVUtbB/view?usp=sharing

Reference

- [1] Large Margin Mechanism and Pseudo Query Set on Cross-Domain Few-Shot Learning
- [2] Cross-Domain Few-Shot Classification via Learned Feature-Wise Transformation