Combat Long-tails in Medical Classification with Relation-aware Consistency and Virtual Features Compensation

Li Pan[1*], Yupei Zhang[2*], Qiushi Yang[3], Tan Li[4], Zhen Chen[5✉]

[1] The Chinese University of Hong Kong
[2] The Centre for Intelligent Multidimensional Data Analysis (CIMDA)
[3] City University of Hong Kong
[4] Department of Computer Science, The Hang Seng University of Hong Kong
[5] Centre for Artificial Intelligence and Robotics (CAIR), HKISI-CAS
zhen.chen@cair-cas.org.hk

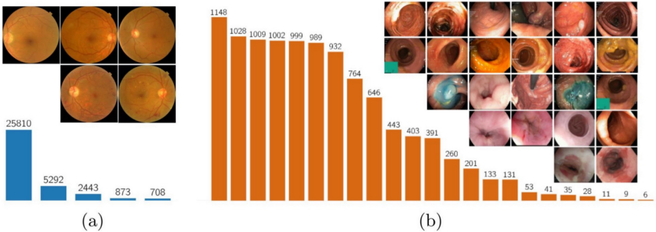
Presenter: Gefei Gu(顧格非)

BACKGROUND

Background

Long-tailed Problem

Scarcity of diseased samples
$\rightarrow$ inherent imbalance datasets
$\rightarrow$ degrade classification performance



(a) DR grading from retinal images; (b) Gastro-intestinal image classification.

Y. Zhang et al., Deep Long-Tailed Learning: A Survey. Arxiv, 2021.
A. Galdran et al., Balanced-MixUp for Highly Imbalanced Medical Image Classification. MICCAI, 2021.

OVERVIEW

Current Approaches

- Reweight the contribution of different class

CROSSENTROPYLOSS

CLASS torch.nn.CrossEntropyLoss(*weight=None*, *size_average=None*, *ignore_index=-100*, *reduce=None*, *reduction='mean'*, *label_smoothing=0.0*) [SOURCE]

This criterion computes the cross entropy loss between input logits and target.

It is useful when training a classification problem with C classes. If provided, the optional argument `weight` should be a 1D *Tensor* assigning weight to each of the classes. This is particularly useful when you have an unbalanced training set.

https://pytorch.org/docs/stable/generated/torch.nn.CrossEntropyLoss.html

Current Approaches

- Rebalance the data distribution (under-sample the head classes, over-sample the tail classes)

Current Approaches

- Rebalance the data distribution (under-sample the head classes, over-sample the tail classes)
- Two-Stage method:
  - Train the model on the entire dataset
  - Fine-tune the classifier using rebalancing techniques

METHODOLOGY

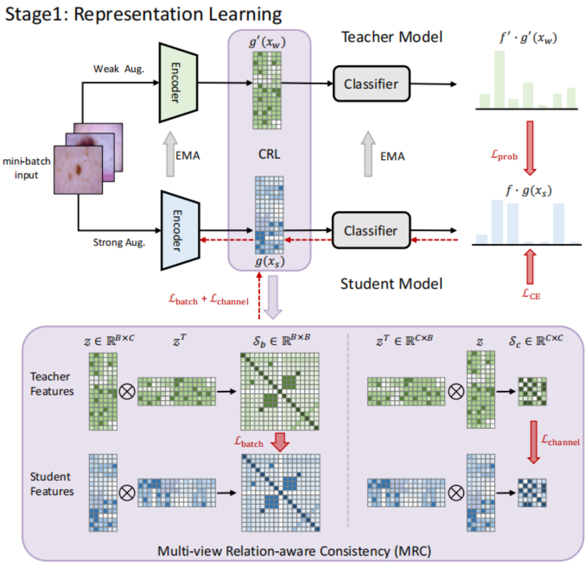
Methodology

- In the first stage, the authors introduce the Multi-view Relation-aware Consistency (MRC) to boost representation learning for the encoder g.

Methodology

- In the first stage, the authors introduce the Multi-view Relation-aware Consistency (MRC) to boost representation learning for the encoder g.
- In the second stage, they proposed Virtual Features Compensation (VFC) to recalibrate the classifier f by generating massive balanced virtual features, which compensates the tails classes without dropping the samples of the head classes.

Stage 1



Stage1: Representation Learning
Teacher Model
Weak Aug.
Encoder
$g'(x_w)$
Classifier
$f' \cdot g'(x_w)$
mini-batch input
EMA
CRL
EMA
$\mathcal{L}_{\text{prob}}$
$f \cdot g(x_s)$
Strong Aug.
Encoder
Classifier
$g(x_s)$
Student Model
$\mathcal{L}_{\text{CE}}$
$\mathcal{L}_{\text{batch}} + \mathcal{L}_{\text{channel}}$
$z \in \mathbb{R}^{B\times C}$
z^T
$\mathcal{S}_b \in \mathbb{R}^{B\times B}$
$z^T \in \mathbb{R}^{C\times B}$
z
$\mathcal{S}_c \in \mathbb{R}^{C\times C}$
Teacher Features
$\mathcal{L}_{\text{batch}}$
$\mathcal{L}_{\text{channel}}$
Student Features
Multi-view Relation-aware Consistency (MRC)

Stage 1

Multi-view Relation-aware Consistency

$$\mathcal{L}_{\text{prob}} = \frac{1}{B}\text{KL}(f \cdot g(\boldsymbol{x}_s), f' \cdot g'(\boldsymbol{x}_w)) \tag{1}$$

$$\mathcal{L}_{\text{batch}} = \frac{1}{B}\|\mathcal{S}_b(g(\boldsymbol{x}_s)) - \mathcal{S}_b(g'(\boldsymbol{x}_w))\|_2 \tag{2}$$

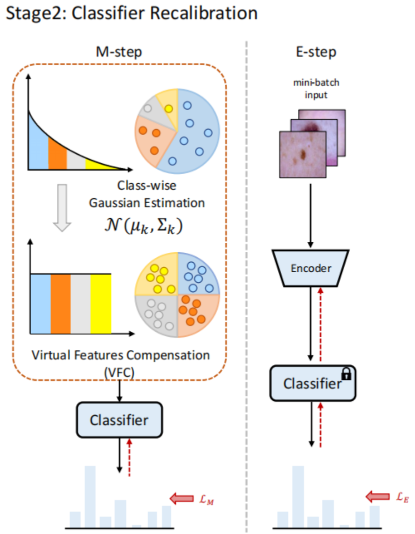
$$\mathcal{L}_{\text{channel}} = \frac{1}{C}\|\mathcal{S}_c(g(\boldsymbol{x}_s)) - \mathcal{S}_c(g'(\boldsymbol{x}_w))\|_2 \tag{3}$$

$$\mathcal{L}_{\text{CE}} = \frac{1}{B}L(f \cdot g(\boldsymbol{x}_w), y) \tag{4}$$

$$\mathcal{L}_{\mathsf{stage1}} = \mathcal{L}_{\mathsf{CE}} + \lambda_1\mathcal{L}_{\mathsf{batch}} + \lambda_2\mathcal{L}_{\mathsf{channel}} + \lambda_3\mathcal{L}_{\mathsf{prob}}$$

In its released code, $\lambda_1, \lambda_2, \lambda_3$ are set as hyper-parameters as 10,10,5.

Stage 2



Expectation-Maximization

E-step:freeze the classifier and train the encoder

$$\mathcal{L}_{\text{stage2}}^{E} = \frac{1}{N}\sum_{\boldsymbol{x}\in X}\frac{(1-(f\cdot g^{I}(\boldsymbol{x})y)^{q})}{q}$$

M-step:freeze the encoder and train the classifier

$$\mathcal{L}_{\text{stage2}}^{M} = \frac{1}{RK}\sum_{k=1}^{K}\sum_{\boldsymbol{v}_i\in V_k} L_{\text{CE}}(f(\boldsymbol{v}_i), y)$$

EXPERIMENTS

Experimental Results

Experiments were performed on two dermatology datasets, on four NVIDIA GeForce GTX 1080 Ti GPUs.

ISIC-2019-LT			
Methods	Acc(%) @ Factor=100	Acc(%) @ Factor=200	Acc(%) @ Factor=500
CE	56.91	53.77	43.89
RS	61.41	55.12	47.76
MixUp [27]	59.85	54.23	43.11
GCE+SR [32]	64.57	58.28	54.36
Seesaw loss [26]	68.82	65.84	62.92
Focal loss [16]	67.54	65.93	61.66
CB loss [8]	67.54	66.70	61.89
FCD [15]	70.15	68.82	63.59
FS [12]	71.97	69.30	65.22
Ours *w/o* MRC	75.04	73.13	70.13
Ours *w/o* VFC	72.91	71.07	67.48
Ours	**77.41**	**75.98**	**74.62**

Comparison on ISIC-2019-LT dataset

ISIC-Archive-LT				
Methods	Head (Acc%)	Medium (Acc%)	Tail (Acc%)	All (Acc%)
CE	**71.31**	49.22	38.17	52.90
RS	70.17	55.29	34.29	53.25
GCE+SR [32]	64.93	57.26	38.22	53.47
Seesaw loss [26]	70.26	55.98	42.14	59.46
Focal loss [16]	69.57	56.21	39.65	57.81
CB loss [8]	64.98	57.01	61.61	61.20
FCD [15]	66.39	61.17	60.54	62.70
FS [12]	68.69	58.74	64.48	63.97
Ours *w/o* MRC	69.06	62.14	65.12	65.44
Ours *w/o* VFC	65.11	62.35	67.30	64.92
Ours	69.71	**63.47**	**70.34**	**67.84**

Comparison on ISIC-Archive-LT dataset

Experimental Results

Significantly promotes the performance of tail classes.

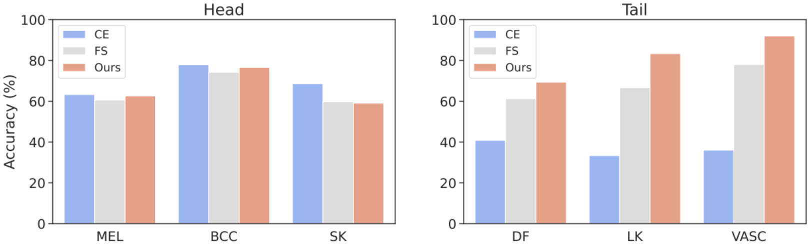


Figure: Comparison of head/tail classes in ISIC-Archive-LT dataset

CONCLUSION

Conclusion

To address the long-tails in computer-aided diagnosis, the authors propose the MRC-VFC framework to improve medical image classification in two stages. MRC introduces multi-view relation-aware consistency. VFC helps to train an unbias classifier by generating massive virtual features. Extensive experiments shows that MRC-VFC outperforms SOTA algorithms remarkably.

THANK YOU!