PCCT: Progressive Class-Center Triplet Loss for Imbalanced Medical Image Classification

Kanghao Chen, Weixian Lei, Shen Zhao, Wei-Shi Zheng, and Ruixuan Wang

***Abstract*—Imbalanced training data in medical image diagnosis is a significant challenge for diagnosing rare diseases. For this purpose, we propose a novel two-stage Progressive Class-Center Triplet (PCCT) framework to overcome the class imbalance issue. In the first stage, PCCT designs a class-balanced triplet loss to coarsely separate distributions of different classes. Triplets are sampled equally for each class at each training iteration, which alleviates the imbalanced data issue and lays solid foundation for the successive stage. In the second stage, PCCT further designs a class-center involved triplet strategy to enable a more compact distribution for each class. The positive and negative samples in each triplet are replaced by their corresponding class centers, which prompts compact class representations and benefits training stability. The idea of class-center involved loss can be extended to the pair-wise ranking loss and the quadruplet loss, which demonstrates the generalization of the proposed framework. Extensive experiments support that the PCCT framework works effectively for medical image classification with imbalanced training images. On four challenging class-imbalanced datasets (two skin datasets Skin7 and Skin 198, one chest X-ray dataset ChestXray-COVID, and one eye dataset Kaggle EyePACs), the proposed approach respectively obtains the mean F1 score 86.20, 65.20, 91.32, and 87.18 over all classes and 81.40, 63.87, 82.62, and 79.09 for rare classes, achieving state-of-the-art performance and outperforming the widely used methods for the class imbalance issue.**

***Index Terms*—Data imbalance, medical image classification, triplet loss.**

Manuscript received 27 July 2022; revised 16 November 2022; accepted 17 January 2023. Date of publication 27 January 2023; date of current version 5 April 2023. This work was supported in part by the National Key Research and Development Program under Grant 2022YFE0112500, in part by the National Natural Science Foundation of China under Grants 62071502, U1811461, and 62101607, and in part by Guangdong Key Research and Development Program under Grant 2020B1111190001. *Kanghao Chen and Weixian Lei contributed equally to this work. (Corresponding authors: Ruixuan Wang; Shen Zhao.)*

Kanghao Chen, Weixian Lei, and Wei-Shi Zheng are with the School of Computer Science and Engineering, Sun Yat-Sen University, Guangzhou 510275, China (e-mail: chenkh25@mail2.sysu.edu.cn; leiwx52@gmail.com; wszheng@ieee.org).

Shen Zhao is with the School of Intelligent Systems Engineering, Sun Yat-Sen University, Shenzhen 518107, China (e-mail: z-s-06@163.com).

Ruixuan Wang is with the School of Computer Science and Engineering, Sun Yat-Sen University, Guangzhou 510275, China, and also with the Department of Network Intelligence, Peng Cheng Laboratory, Shenzhen 510006, China (e-mail: wangruix5@mail.sysu.edu.cn).

Digital Object Identifier 10.1109/JBHI.2023.3240136

I. Introduction

CLASS imbalance issue is ubiquitous in medical diagnosis [1] because large-scale clinical datasets often exhibit imbalanced class distributions. For example, in clinical diagnosis, the data is by nature heavily imbalanced [2] because common diseases occur more frequently than rare disease. This raises a major challenge for modern deep learning models because most of them assume balanced class distributions in the training dataset. When presented with an imbalanced dataset, the training procedure is dominated by frequent classes, and the trained model tends to perform better on these frequent classes but significantly worse on infrequent classes [3].

Many approaches attempt to solve the class imbalance issue. For example, the re-sampling [4] strategy can be applied to over-sample the limited data from infrequent classes or under-sample the data from frequent classes to balance training data across classes. The re-weighting [5] strategy sets larger weights to the loss terms related to infrequent classes, which makes balanced loss terms across classes. Still relevant to modification of training loss, the traditional margin-based loss may be refined by setting smaller margin for frequent classes and larger margin for infrequent classes to alleviate the class-imbalance issue [6]. Another approach [7] adaptively sets a higher weight for the sample that is difficult to recognize. Besides the re- balancing strategies mentioned above, other strategies have also been proposed by improving the representation ability of the deep neural network. These can be achieved by class-imbalanced representation learning, such as transfer learning [8], semi-supervised and self-supervised learning [9]. The above-mentioned strategies can be combined with each other, for example, by first performing representation learning of the feature extractor and then applying re-balancing strategy to the model output side [10] or the input side [11], the classification performance of the rare classes can be improved.

However, although these strategies alleviate the class imbalance issue to some extent, their adverse effects should also be considered. For example, re-sampling has the risks of over-fitting the small-sample classes and under-fitting the larger-sample classes, and re-weighting may distort the original distributions by directly changing or over-inverting the data frequency of infrequent classes. This could unexpectedly damage the overall representation ability of the learned features. Also, the combination of representation learning [8] and re-balancing strategies [10] might also suffer from higher sensitivity to hyper-parameters or higher complexity in the

training procedure [12]. The other strategy, transfer learning, may require designing complicated modules that require large memory consumption during feature transfer [11].

Different from the above model training approaches, triplet loss [13], [14], [15] is a popular scheme that has the potential to solve the imbalanced issue by extracting discriminative feature representations without tedious hyper-parameter tuning and module training [16], [17]. Triplet loss flexibly acquires anchor, positive, and negative samples where the anchor and positive are from the same class while the negative is from a different class. It then forces the features to be similar (or dissimilar) for samples from the same (or different) class and thus helps the CNN extractors to extract more discriminative features. Triplet loss and its variants can be applied to various applications such as face recognition [13], person re-identification [15] and 3D shape retrieval [16]. For medical applications, methods based on triplet loss have been validated in wireless capsule endoscopy polyp detection [18] and brain tumor classification [19]. However, the potential of triplet losses in solving the class imbalance issue has not been fully exploited. Simply applying the triplet loss to class-imbalanced image classification may encounter the following issues. First, if the anchors, positives, and negatives in the triplets are not elaborately chosen, the triplets whose anchors come from frequent classes will be much more than triplets whose anchors come from the infrequent class.

In this case, triplets are not equally sampled and the frequent classes still dominate the model training, which results in worse classification performance on infrequent classes. Second, when constructing the triplets, anchor and positive samples may be mapped far away in the embedding space particularly for the frequent classes. This may be due to the fact that intra-class visual variations are large in frequent classes, and it may cause instability during triplet-based model training and harm the classification performance. In this study, by extending the triplet loss, we propose a novel two-stage Progressive Class-Center Triplet (PCCT) framework as a stable triplet-based model training strategy to solve the class imbalanced issue for diagnosing rare diseases. The two stages of our PCCT are designed particularly to handle the above-mentioned two potential issues when applying triplet loss to the class-imbalanced classification task. In the first stage, a class-balanced triplet sampling strategy is designed to sample triplet equally for each class, i.e., in each class-balanced triplet batch, the numbers of triplets whose anchors are from different classes are the same. In this way, the number of training data (i.e., triplets) relevant to infrequent classes can be largely increased compared to the number of individual images in these classes, i.e., the triplet data can be balanced between classes. In the second stage, a class-center involved triplet is designed to deal with the problem that anchor and positive samples may be mapped far away. Here the positive and negative samples in each original triplet are replaced by their corresponding class centers. This avoids the cases where the anchor and positive samples are far away and can help establish compact distribution for each class, which keeps the training more stable particularly for datasets with larger intra-class variance. Based on the class-center triplet loss, we further introduce trainable class centers, where class centers are part of trainable model weights and therefore avoids the frequent computation of class centers with all training samples during model training. Different from popular two-stage techniques which mainly focus on the classifier head (e.g., LDAM [6] and Decouple [10]), the proposed two-stage progressive framework boosts the representation of feature extractor in a coarse-to-fine triplet-based model training manner and effectively alleviate the class imbalance issue. The main contributions of this study are summarized as follows.

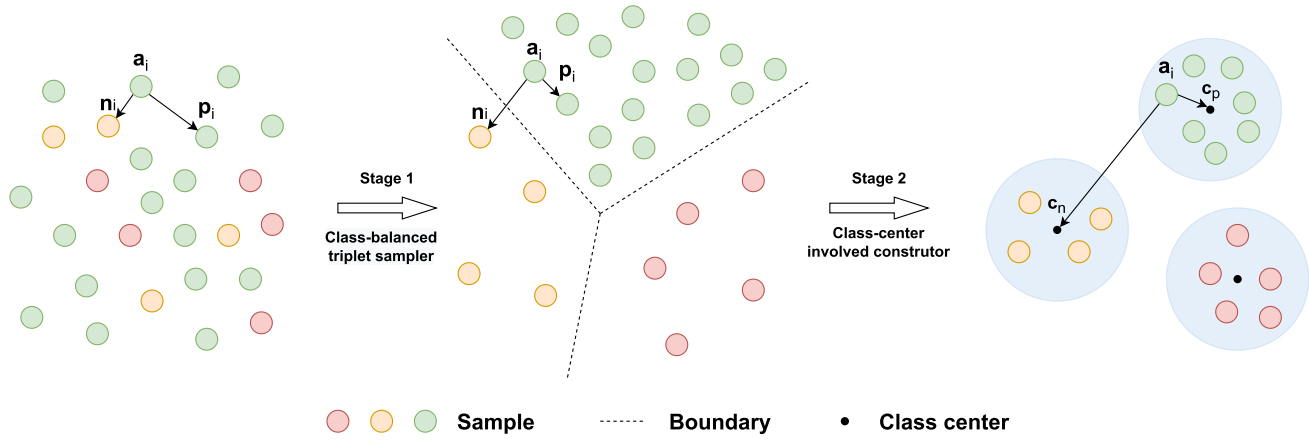
- We propose a novel two-stage Progressive Class-Center Triplet (PCCT) framework to capture discriminative and class-compact representation, which is the first triplet framework for solving the class imbalance issue. This framework is also proved to be extensible to other metric learning schemes such as pair-wise ranking loss and quadruplet loss.
- For the first time, we propose a set of new strategies associated with triplet-based training, including the class-balanced triplet sampler, the class-center involved triplet loss, and the trainable class centers. These strategies help capture discriminative, class-compact, and effective feature representation, and are beneficial for stabilizing triplet-based training and improving classification performance on class-imbalanced datasets.
- The strategies for triplet-based model training are extensively evaluated on multiple imbalanced medical image datasets of different body parts, diseases, and imbalance ratios, which demonstrates the effectiveness of the PCCT framework particularly for rare disease diagnosis.

Note that this work is an extension of the previous conference publication [20] in the following aspects.

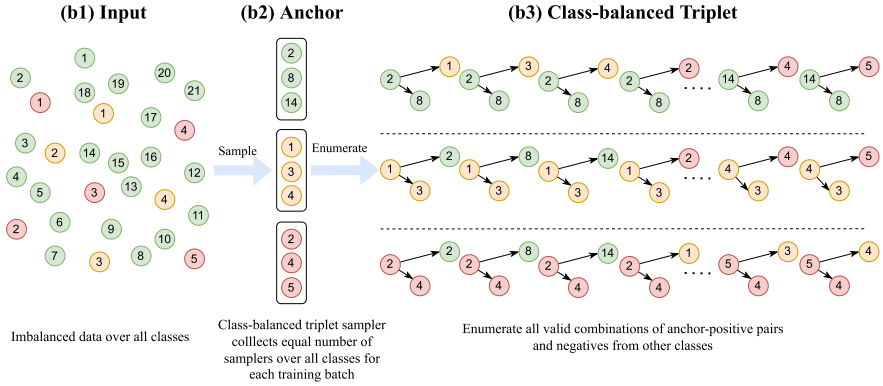
1) We delve into the two-stage training process of our PCCT framework. More detailed analyses of the relationship between the two stages and extensive experiments are performed, clearly demonstrating the effectiveness of the novel training process in the imbalanced medical diagnosis.
2) We propose another strategy for class-center involved triplet loss, which directly trains class centers together with the feature extractor. This requires much less computation cost during training but equivalent classification performance, which is referred to as Efficient-PCCT.
3) We have validated that the class-center involved loss can be extended to pair-wise ranking loss and the quadruplet loss. Corresponding experimental evaluations show that the class-center involved metric learning outperforms the original learning strategies.
4) More extensive evaluations have been included, not only on the two skin image datasets but also on the new chest X-ray dataset and the Kaggle EyePACs dataset.

II. Methods

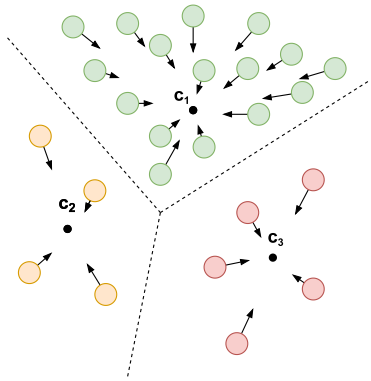
As demonstrated in Fig. 1(a), our Progressive Class-Center Triplet (PCCT) is designed as a two-stage triplet-based model training framework to effectively alleviate the imbalanced issue in medical diagnosis and accurately diagnose both common



(a) Overview of our PCCT procedure.



(b) Class-balanced triplet constructor for alleviating the class imbalance issue.



(c) Class-center involved triplets for encouraging compact representation for each class.

Fig. 1. Our proposed PCCT framework uses a two-stage training procedure to extract discriminative features for imbalanced medical classification. As Fig. 1(a) demonstrates, our PCCT first trains the model by the class-balanced triplet sampler to capture a coarse class center for each class, and then fine-tune the model using the class-center involved triple loss to capture more compact class distributions. Fig. 1(b) demonstrates how the class-balanced triplet sampler constructs triplets, with each triplet composed of an anchor, a positive, and a negative sample. Fig. 1(c) demonstrates the effect of class-center involved triple loss, i.e., achieving compact distribution for each class.

and rare diseases. In the first stage, we design a class-balanced triplet sampler to obtain a batch of triplets whose anchors from different classes are balanced. This strategy samples triplets equally for each class and helps to alleviate the imbalanced issue during training the feature extractor. Furthermore, this training stage helps the feature extractor to capture a coarse class center for each class. In the second stage, we design a class-center involved triplet constructor to obtain triplets whose positives and negatives are the class centers. This strategy enforces the feature representation of each data to be closer to the corresponding class center and therefore helps the distribution of each class to be more compact in feature space, which stabilizes the triplet-based model training procedure. To further explore the two-stage framework, we also show that the class centers can either be directly calculated with training samples of the class during model training or learned as part of model parameters. Moreover, the two-stage framework is proved to be extensible to other metric learning losses (e.g., pair-wise and quadruplet losses), which demonstrates the generality of the proposed method.

A. Triplet Loss With Class-Balanced Triplet Sampler

In the first stage, we propose the class-balanced triplet sampler to obtain batches of triplets whose anchors from different classes are balanced, which alleviates the imbalanced issue and trains a discriminative feature extractor to capture coarse class centers. In order to better describe our method, we first briefly retrospect the classical triplet loss. Each triplet is composed of an anchor, a positive and a negative, where the anchor and the positive are sampled from the same class, and the negative is sampled from other classes. Then the triplet loss is adopted to help satisfy the condition that the distance between the anchor and the positive is closer than the distance between the anchor and negative by a margin α in the feature space, i.e.,

$$\|\mathbf{f}(\mathbf{a}_i;\theta)-\mathbf{f}(\mathbf{p}_i;\theta)\|+\alpha<\|\mathbf{f}(\mathbf{a}_i;\theta)-\mathbf{f}(\mathbf{n}_i;\theta)\|, \quad (1)$$

where $\mathbf{a}_i$, $\mathbf{p}_i$ and $\mathbf{n}_i$ respectively denote an anchor, positive and negative. $\mathbf{f}(\cdot;\theta)$ means the CNN-based feature extractor (e.g., ResNet-50) to be learned, θ denotes the trainable parameters in the CNN. $\|\cdot\|$ can be any L_p norm ($p=2$ by default), and α is

the margin in the inequality constraint (1), which is set to 0.5 by default. In this way, the triplet loss for the CNN model can be defined as

$$l_t(\mathbf{a}_i, \mathbf{p}_i, \mathbf{n}_i; \boldsymbol{\theta}) = [\|\mathbf{f}(\mathbf{a}_i; \boldsymbol{\theta}) - \mathbf{f}(\mathbf{p}_i; \boldsymbol{\theta})\| + \alpha - \|\mathbf{f}(\mathbf{a}_i; \boldsymbol{\theta}) - \mathbf{f}(\mathbf{n}_i; \boldsymbol{\theta})\|]_+ , \quad (2)$$

with $[d]_+ = \max(0, d)$ denoting the hinge loss. The total loss of a batch can be simply calculated by summation over the samples, i.e., $L(\boldsymbol{\theta}) = \frac{1}{N}\sum_{i=1}^{N} l_t(\mathbf{a}_i, \mathbf{p}_i, \mathbf{n}_i; \boldsymbol{\theta})$. $L(\boldsymbol{\theta})$ can be combined with the cross-entropy loss to optimize the CNN model simultaneously. Although the triplet loss can help train a discriminative feature extractor, as mentioned in the Introduction section, traditional triplet losses often ignore class distributions when forming the triplets, i.e., triplets are not sampled equally and the frequent class still dominates the model training in class imbalanced datasets.

In our work, a class-balanced triplet sampler is designed to alleviate the imbalanced issue by constructing batches of triplets whose anchors from different classes are balanced. It first collects equivalent number of samples from all classes to form a training batch. For example, as shown in Fig. 1(b1)(b2), if the batch size *N* is 9 and the class number *C* is 3, then N_i=3 samples are selected from each class. Then, for each class in the batch, all N_i samples of this class are used as anchors to construct anchor-positive pairs, i.e., all possible combinations of the anchor-positive pairs within the batch are considered. Next, for each anchor-positive pair, all possible random-hard [13] negative samples (i.e., all samples from the other classes that make the loss ((2)) larger than 0) are chosen to form as many triplets as possible, as shown by the circles of different colors (classes) in Fig. 1(b3). This combination of anchor-positive pairs and negatives takes full advantages of all samples to form triplets while keeping the balance of the triplets among classes, i.e., triplets whose anchors are of different classes are approximately the same. This alleviates the class imbalance issue in triplet construction. For example, although there are more green samples than the red samples in Fig. 1(b1) (i.e., class imbalance), triplets whose anchors are of different colors (classes) are kept approximately the same in Fig. 1(b3) through the class-balanced triplet sampling strategy. In this way, our class-balanced sampler generates class-balanced batches during model training, i.e., the number of anchors from different classes remains equivalent in each batch in the training process, therefore helping to alleviate the class imbalance issue.

Another advantage of the first training stage is that it can capture a coarse class center for each class. By training the feature extractor $\boldsymbol{\theta}$ with the class-balanced triplets, the distances between the samples from the same class are smaller than those between the samples from different classes, i.e., the embedding features of the samples within one class will be closer. In this way, the averaged sample features, i.e., the coarse class centers, of each class would be more representative of this class. These coarse class centers can help the feature extractor in the next stage to capture a more compact feature distribution for each class, which helps to stabilize the training process. Without the first-stage training process, randomly initialized parameters of the feature extractor would probably cause the distributions to spread more or less randomly and heavily overlap across classes in the feature space especially at the early training stage. Such frequently and largely changed class distribution over training epochs would cause the training instability in the second stage when class centers are involved. Therefore, the first stage sets the foundation for the second stage to converge more stably.

B. Class-Center Involved Triplet Constructor

After training the CNN feature extractor model in the first stage, a class-center involved triplet loss is designed to further improve the class-imbalanced classification performance in the second stage. In this stage, the class-center involved triplet loss is designed to enforce the feature representation of each data to be closer to the corresponding class center in the feature space, therefore prompting the overall class-level distribution to be more compact in the feature space. This design is motivated by the fact that for a frequent (larger-sample) class in a large-scale medical dataset, images could have large variations in visual appearance even if they are from the same class. This would lead to a spreading distribution in a relatively large region in the feature space for the frequent classes, i.e., the anchor and positive samples can be far away from each other in the feature space. This can cause instability during training and harm the classification performance in traditional triplets. The class-center involved triplet is designed to deal with the problem and improve training stability.

To improve the optimization process, we propose modifying the original triplet loss with class centers to consider the global information on distributions of all classes. As demonstrated in Fig. 1(c), the modified triple loss is designed as

$$l_c(\mathbf{a}_i, \mathbf{p}_i, \mathbf{n}_i; \boldsymbol{\theta}_t) = [\|\mathbf{f}(\mathbf{a}_i; \boldsymbol{\theta}_t) - \mathbf{c}(\mathbf{a}_i; \boldsymbol{\theta}_{t-1})\| + \alpha - \|\mathbf{f}(\mathbf{a}_i; \boldsymbol{\theta}_t) - \mathbf{c}(\mathbf{n}_i; \boldsymbol{\theta}_{t-1})\|]_+ . \quad (3)$$

Similar to Eq. 2, $\mathbf{f}(.; \boldsymbol{\theta})$ denotes a CNN feature extractor with parameters $\boldsymbol{\theta}$. However, the extracted feature $\mathbf{f}(\mathbf{p}_i; \boldsymbol{\theta})$ and $\mathbf{f}(\mathbf{n}_i; \boldsymbol{\theta})$ of the positive and negative samples in Eq. 2 are respectively the corresponding class centers $\mathbf{c}(\mathbf{a}_i; \boldsymbol{\theta}_{t-1})$ and $\mathbf{c}(\mathbf{n}_i; \boldsymbol{\theta}_{t-1})$. The class-center involved triplets are constructed as follows. First, at the beginning of each training epoch, all training samples are fed into the feature extractor (with parameters $\boldsymbol{\theta}$) to obtain their feature representations. Then, the feature representations of all training samples from the same class are averaged to obtain the class center for each class. Next, at each training iteration, each sample in the batch is regarded as the anchor, and the class center of the same class is regarded as the positive for the anchor. Subsequently, the distance of the anchor to all class centers are calculated, and those centers satisfying the condition $l_c(\mathbf{a}_i, \mathbf{p}_i, \mathbf{n}_i; \boldsymbol{\theta}_t) > 0$ (from Eq. 3) are selected as negatives, i.e., class centers of the other different classes which are nearer to the anchor than the positive by a distance margin α are regarded as negatives and used for constructing triplets. The triplets are used to calculate the class-center involved triplet loss for training the feature extractor. Lastly, the class centers are updated after each epoch.

The designed class-center involved triplet construction not only benefits triplet-based model training, but also makes full use of the training samples. From the aspect of triplet-based model training, since the class centers, i.e., the averaged features all samples of this class, are used to construct triplets, the outlier samples of the frequent classes can be automatically avoided in triplet construction. The triplets constructed in this manner tend to satisfy Eq. 3, which are beneficial for triplet-based model training. Also, by using class centers as positives and negatives, the circumstance that anchor and positive samples are mapped far away is avoided, which stabilizes the training when sample features show large variety in the frequent class. Also particularly after certain epochs of training, the locations of the class centers in the feature space would not show drastic changes, which helps to stabilize the training. From the aspect of sample exploitation, for a batch of samples, all of them are exhaustively used as the anchors. For each anchor sample, all negative class centers that contributes to non-zero losses (i.e., the values of Eq. 3 are larger than 0) are selected to form triplets with the anchor. Thus, triplets beneficial for training are constructed for each sample, i.e., all samples are sufficiently leveraged to construct triplets. In comparison, the triplet center loss for object retrieval in [16] uses only the nearest negative center for each anchor. In conclusion, our PCCT fully considers the global distribution of different classes and provides effective and stable training of the CNN feature extractor. During the testing procedure, since the CNN feature extractor has been well trained, the class centers are representative of all classes.

Thus, we can simply adopt the nearest class center method to make a prediction for any new test sample, i.e., we can classify the test sample to the class whose center is closest to the test sample in the feature space.

C. Extensions of Class-Center Based Triplet Loss

Our two-stage PCCT can be extended to the pair-wise loss [21] and the quadruplet loss [22]. Pair-wise loss is one of basic methods in metric learning, and quadruplet loss has been shown to cause larger inter-class variation and smaller intra-class variation in the feature space compared to the triplet loss, which can obtain a better generalization ability in person ReID [23]. In order to better describe the extended methods, we first formalize the commonly used pair-wise loss and quadruplet loss. The pair-wise ranking loss enforces the paired samples from the same class to be close to each other and the paired samples from different classes to be further apart. The pair-wise ranking loss l_p is defined with paired samples of same or different classes,

$$\begin{aligned} & l_p(\mathbf{a}_i, \mathbf{b}_i; \boldsymbol{\theta}) \\ = & \mathbb{1}(\mathbf{a}_i, \mathbf{b}_i) \|\mathbf{f}(\mathbf{a}_i; \boldsymbol{\theta}) - \mathbf{f}(\mathbf{b}_i; \boldsymbol{\theta})\| \\ & + (1 - \mathbb{1}(\mathbf{a}_i, \mathbf{b}_i))[\alpha - \|\mathbf{f}(\mathbf{a}_i; \boldsymbol{\theta}) - \mathbf{f}(\mathbf{b}_i; \boldsymbol{\theta})\|]_+ , \end{aligned} \tag{4}$$

where the indicator function $\mathbb{1}(\mathbf{a}_i, \mathbf{b}_i)$ is 1 if $\mathbf{a}_i$ and $\mathbf{b}_i$ belong to the same class, and 0 otherwise.

Compared to triplet loss, the quadruplet loss additionally enforces that the distance between two samples of the same class is smaller than that between two samples from another two classes. The quadruplet loss is defined based on quadruple samples which include one anchor and one positive from the same class, and two negative samples ($\mathbf{n}_{i,1}$ and $\mathbf{n}_{i,2}$) coming from two other different classes,

$$\begin{aligned} l_q(\mathbf{a}_i, \mathbf{p}_i, \mathbf{n}_{i,1}, \mathbf{n}_{i,2}; \boldsymbol{\theta}) = & [\|\mathbf{f}(\mathbf{a}_i; \boldsymbol{\theta}) - \mathbf{f}(\mathbf{p}_i; \boldsymbol{\theta})\| + \alpha \\ & - \|\mathbf{f}(\mathbf{a}_i; \boldsymbol{\theta}) - \mathbf{f}(\mathbf{n}_{i,1}; \boldsymbol{\theta})\|]_+ \\ & + [\|\mathbf{f}(\mathbf{a}_i; \boldsymbol{\theta}) - \mathbf{f}(\mathbf{p}_i; \boldsymbol{\theta})\| + \beta \\ & - \|\mathbf{f}(\mathbf{n}_{i,1}; \boldsymbol{\theta}) - \mathbf{f}(\mathbf{n}_{i,2}; \boldsymbol{\theta})\|]_+ , \end{aligned} \tag{5}$$

where β is another constant that is normally smaller than α.

For extension, the class-center based pair-wise ranking loss and quadruplet loss can be obtained by simply using class centers as corresponding positive and negative samples, i.e.,

$$\begin{aligned} & l_{pc}(\mathbf{a}_i, \mathbf{b}_i; \boldsymbol{\theta}_t) \\ = & \mathbb{1}(\mathbf{a}_i, \mathbf{b}_i) \|\mathbf{f}(\mathbf{a}_i; \boldsymbol{\theta}_t) - \mathbf{c}(\mathbf{b}_i; \boldsymbol{\theta}_{t-1})\| \\ & + (1 - \mathbb{1}(\mathbf{a}_i, \mathbf{b}_i))[\alpha - \|\mathbf{f}(\mathbf{a}_i; \boldsymbol{\theta}_t) - \mathbf{c}(\mathbf{b}_i; \boldsymbol{\theta}_{t-1})\|]_+ , \end{aligned} \tag{6}$$

$$\begin{aligned} & l_{qc}(\mathbf{a}_i, \mathbf{p}_i, \mathbf{n}_{i,1}, \mathbf{n}_{i,2}; \boldsymbol{\theta}_t) \\ = & [\|\mathbf{f}(\mathbf{a}_i; \boldsymbol{\theta}_t) - \mathbf{c}(\mathbf{p}_i; \boldsymbol{\theta}_{t-1})\| + \alpha \\ & - \|\mathbf{f}(\mathbf{a}_i; \boldsymbol{\theta}_t) - \mathbf{c}(\mathbf{n}_{i,1}; \boldsymbol{\theta}_{t-1})\|]_+ \\ & + [\|\mathbf{f}(\mathbf{a}_i; \boldsymbol{\theta}_t) - \mathbf{c}(\mathbf{p}_i; \boldsymbol{\theta}_{t-1})\| + \beta \\ & - \|\mathbf{c}(\mathbf{n}_{i,1}; \boldsymbol{\theta}_{t-1}) - \mathbf{c}(\mathbf{n}_{i,2}; \boldsymbol{\theta}_{t-1})\|]_+ . \end{aligned} \tag{7}$$

Once the feature extractor is well trained by the pair-wise ranking loss or the quadruplet loss, it can be used for classification of any test sample using the nearest class-center method as mentioned above, which is accessible in the testing stage.

D. Trainable Class Centers

Besides the extension to other losses, we also carry out another exploration around the topic of the two-stage triplet-based model training pipeline. In this exploration, we consider the class centers in the second stage of the proposed PCCT framework as trainable model parameters to reduce the computational cost and speed up the training with the two-stage PCCT. In other words, the class center (i.e., $\mathbf{c}(\cdot; \boldsymbol{\theta}_{t-1})$) in Eq. 3 become trainable class centers. The idea of trainable centers is motivated by the observation that the calculation of the class centers would take up a lot of computation, probably because this procedure needs to fetch all training samples and average features for each class.

Formally, we define the class centers $\mathbf{C} = \{\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_K\}$ as part of model parameters, where K represents the total number of classes. The trainable centers are also implemented with the designed two-stage framework. For the first stage, the same class-balanced triplet sampler is used to train the embedding network (with network parameters $\boldsymbol{\theta}$) to extract compact feature distribution for each class. Then, for the second stage, the embedding network is initialized with that trained in the first stage. What is different, the trainable center strategy initializes the centers for all classes as trainable vectors, which are parts of the model parameters that can be optimized with the parameters

TABLE I
THE STATISTICS OF THREE MEDICAL IMAGE DATASETS

Dataset	Class number	Image number in largest class	Image number in smallest class
Skin7 [1], [26]	7	6705	115
Skin198 [27]	198	60	10
ChestXray-COVID [28]	3	8851	133
Kaggle EyePACs [29]	2	22629	5471

TABLE II
THE BATCH SIZE DURING TRAINING FOR EACH DATASET

	Skin7	Skin198	ChestXray-COVID	Kaggle EyePACs
First stage	10 per class	5 per class	5 per class	10 per class
Second stage	16	32	8	32

in the network in an end-to-end manner. Then, following the two-stage training workflow, each sample in a training batch is regarded as the anchor, the class center with the same label is regarded as the positive, and all centers with different labels satisfying Eq. 3 are regarded as the negatives. In this way, the gradient back-propagation procedure with the loss function updates the trainable class centers $\mathbf{C}$ together with the feature extractor parameters θ. This procedure avoids passing all samples through the feature extractor to calculate the averaged class centers after each training epoch, which avoids the time-consuming procedure during training. Since the trainable centers contain relatively fewer ($class\_number \times feature\_vector\_length$) parameters, the additional computational cost caused by the trainable class centers during training is negligible compared with the computed class centers. We propose to introduce this strategy into the original PCCT and called the new version Efficient-PCCT.

A recently study introduces trainable class centers [24] to maximize a Gaussian affinity objective with hyper-parameters to be tuned and approximately equal distance between class centers to be enforced. Differently, our trainable class centers can be simply obtained by minimizing the proposed class-center involved triplet loss without extra constraints.

III. EXPERIMENTS AND DISCUSSIONS

A. Experiment Settings

We choose four challenging datasets (Table I) for imbalanced medical diagnosis to evaluate the proposed PCCT method. All four datasets include frequent and infrequent classes, with varying levels of class imbalance. The dermoscopy dataset Skin7 contains 7 categories with an imbalance ratio of 58.3 (i.e., 6705/115). Another skin image dataset Skin198 contains 198 categories with an imbalance ratio of 6.0. The X-ray dataset ChestXray-COVID contains 3 categories with an imbalance ratio of 66.5, including 8,851 "Normal" images, 1,000 "Pneumonia" images and 133 "COVID" images. The Kaggle EyePACs consists of 35,126 images with labels. Following the relevant study [25], we recast the 5-class classification task as binary classification. The recasted EyePACs contains 2 categories with an imbalance ratio of 4.14, with the frequent class (healthy) containing 22629 images and the infrequent class containing 5471 images in the training set. During training, all Skin7, Skin198 and EyePACs images are resized to 300×300 pixels and then randomly cropped to 224×224 pixels, while ChestXray-COVID images are resized to 512 followed by a random horizontal flip. The batch size setting (Table II) is chosen for better convergence. For our PCCT, the last output layer of the original CNN is replaced by a new fully-connected layer whose output is a 128-dimensional feature vector. We use Adam optimizer to train the CNN feature extractor. The learning rate, β_1, and β_2 are respectively set as 0.0001, 0.9, and 0.99. α is set to 0.5. Both two stages in PCCT are trained for 200 epochs to ensure the convergence of the CNN model.

For evaluation, the proposed PCCT is compared with multiple relevant baselines with the same backbone (ResNet-50) feature extractor. Our PCCT has been extensively evaluated using three evaluation metrics, i.e., average precision ('MCP'), average recall ('MCR'), and average F1 score ('MF1') over classes. MF1 takes both MCP and MCR into account, which is considered as a more comprehensive measurement to evaluate the model. Standard 5-fold cross-validation is performed to obtain comprehensive performance comparison. The means and standard deviations (in bracket) of the three measurements over five folds are reported in each evaluation. All the measurements are reported in the form of percentage (i.e., %), which is omitted for brevity.

B. Effectiveness of the Triplet-Based Approach

The effectiveness of the proposed PCCT is evaluated by comparing with several popular class imbalance training strategies, including the class re-weighting strategy ('WCE') [5], the oversampling strategy ('OCE'), the focal loss ('WFCE') [7]. All above methods are based on the cross-entorpy loss, so the abbreviations are annotated with 'BCE'. We also compare our method with the state-of-the-art two-stage decoupling method ('TSD') [10]. The basic cross-entropy loss ('BCE') [30] without any class re-balancing strategy is also included for comparison. In training, the batch size is set to 32 for all baselines and the same training and evaluation protocols are used as for the proposed approach.

Table III and Table IV show that our PCCT outperforms the compared methods on all the datasets on both overall performance and infrequent class classification performances. As shown in the last two rows of Table III, our PCCT, as well as Efficient PCCT, show consistent superior performance with all three metrics. For example, on the Skin7 dataset, PCCT achieves the mean F1 score of 86.2%, which is better than all the other methods. Also, the mean F1 performance over all classes shows improvements on the Skin198, ChestXray-COVID, and Kaggle EyePACs datasets. Wilcoxon Rank test demonstrates our PCCT significantly outperforms the compared methods (p-values $<$ 0.05). Also, the performance of our PCCT is comparable to the state-of-the-art approach TSD (there is no significant difference between PCCT and TSD). For the performance of PCCT on infrequent classes, we compare the performance of all methods on the most infrequent class in the four datasets. For Skin7, ChestXray-COVID, and Kaggle EyePACs, the most infrequent

TABLE III
THE OVERALL PERFORMANCE OF PCCT IS AMONG THE BEST COMPARED WITH THE STATE-OF-THE-ART METHODS IN SKIN7, SKIN198, CHESTXRAY-COVID, AND KAGGLE EYEPACS

Methods	Skin 7			Skin 198			ChestXray-COVID			Kaggle EyePACs		
	MF1	MCP	MCR	MF1	MCP	MCR	MF1	MCP	MCR	MF1	MCP	MCR
BCE [30] (2016)	83.65 (1.52)	86.96 (1.96)	81.15 (1.62)	51.91 (1.10)	56.41 (1.27)	52.12 (1.14)	82.37 (2.97)	91.35 (3.17)	76.95 (3.35)	86.52 (1.28)	88.57 (1.48)	84.51 (1.36)
WCE [5] (2019)	82.45 (1.31)	83.35 (1.79)	82.06 (1.47)	60.21 (1.36)	64.82 (1.34)	60.23 (1.12)	82.18 (5.44)	93.59 (1.99)	76.24 (6.71)	86.02 (3.43)	87.54 (2.35)	84.97 (1.48)
OCE [31] (2019)	83.53 (1.33)	87.26 (1.27)	80.81 (1.39)	59.77 (1.89)	64.87 (2.06)	59.34 (1.87)	84.16 (4.51)	92.21 (3.55)	78.77 (5.11)	85.58 (0.99)	86.13 (1.80)	85.05 (0.64)
WFCE [7] (2017)	83.52 (1.63)	86.43 (1.34)	81.25 (1.78)	53.28 (2.65)	58.31 (2.77)	53.34 (2.58)	83.77 (3.71)	91.31 (3.08)	79.32 (4.15)	86.99 (1.65)	89.56 (1.76)	84.08 (1.98)
TSD [10] (2020)	86.00 (1.02)	87.74 (1.31)	84.63 (1.19)	64.23 (1.54)	67.10 (1.90)	65.62 (1.56)	91.13 (0.76)	94.22 (1.69)	88.47 (1.28)	86.91 (1.91)	**90.09** (2.98)	84.51 (1.31)
Triplet-ratio loss [32] (2022)	84.85 (1.40)	87.98 (2.18)	82.37 (1.31)	63.91 (2.11)	66.81 (2.18)	64.42 (2.00)	83.85 (1.64)	90.79 (3.67)	78.94 (3.35)	86.93 (0.73)	87.52 (1.50)	**85.62** (0.98)
PCCT	86.20 (1.07)	87.77 (1.54)	**84.98** (0.75)	**65.20** (1.49)	68.40 (1.36)	**66.02** (1.50)	**91.32** (0.73)	**94.67** (0.66)	**88.49** (1.35)	87.18 (1.21)	89.86 (1.11)	85.08 (1.30)
Efficient-PCCT	**86.36** (0.95)	**88.97** (1.51)	84.32 (0.70)	64.51 (1.65)	**68.76** (1.80)	64.74 (1.84)	89.91 (1.44)	93.57 (2.42)	86.95 (2.59)	**87.44** (1.06)	90.05 (1.79)	85.38 (1.11)

TABLE IV
THE PERFORMANCE OF PCCT ON INFREQUENT CLASSES IS AMONG THE BEST COMPARED WITH THE STATE-OF-THE-ART METHODS IN SKIN7, SKIN198, CHESTXRAY-COVID, AND KAGGLE EYEPACS

Methods	Skin 7			Skin 198			ChestXray-COVID			Kaggle EyePACs		
	MF1	MCP	MCR	MF1	MCP	MCR	MF1	MCP	MCR	MF1	MCP	MCR
BCE [30] (2016)	73.67 (3.62)	79.03 (0.76)	69.39 (6.24)	18.59 (2.43)	24.22 (3.00)	16.67 (2.78)	71.24 (6.65)	95.11 (5.02)	57.26 (7.86)	78.84 (3.65)	85.61 (5.65)	71.07 (5.73)
WCE [5] (2019)	77.96 (5.31)	87.18 (2.47)	70.83 (7.64)	53.37 (1.99)	65.21 (2.52)	49.79 (2.68)	69.85 (16.84)	**97.60** (5.37)	57.15 (20.19)	76.88 (3.73)	78.56 (2.56)	72.70 (1.54)
OCE [31] (2019)	74.05 (8.91)	84.93 (5.16)	66.17 (11.81)	56.41 (3.55)	66.46 (4.25)	53.42 (3.17)	72.72 (13.09)	91.52 (9.33)	61.00 (14.90)	76.79 (2.23)	78.36 (5.36)	72.89 (5.07)
WFCE [7] (2017)	76.21 (4.94)	84.96 (3.61)	69.35 (6.62)	20.36 (2.08)	26.83 (2.74)	17.99 (2.21)	72.52 (10.91)	93.64 (7.51)	60.31 (13.58)	78.69 (3.75)	**87.50** (3.69)	71.50 (4.04)
TSD [10] (2020)	80.73 (7.76)	87.11 (7.51)	75.65 (9.76)	62.89 (3.57)	66.65 (4.12)	65.10 (3.67)	**83.47** (1.69)	88.13 (2.37)	**79.40** (3.26)	78.59 (3.55)	86.93 (6.88)	71.71 (1.85)
Triplet-ratio loss [32] (2022)	80.12 (5.91)	86.35 (3.33)	73.04 (8.87)	64.56 (3.45)	65.71 (1.11)	64.42 (1.36)	74.88 (1.07)	83.34 (5.64)	68.40 (4.99)	78.61 (4.20)	83.33 (2.30)	73.44 (5.57)
PCCT	**81.40** (5.48)	84.16 (2.67)	**79.13** (8.43)	63.87 (3.21)	67.43 (2.74)	**65.33** (3.88)	82.62 (1.52)	88.11 (1.89)	77.90 (3.28)	79.09 (1.78)	86.18 (4.12)	73.07 (3.49)
Efficient-PCCT	81.11 (5.59)	**90.32** (2.42)	73.91 (8.25)	**64.93** (3.64)	**71.44** (3.43)	64.21 (4.04)	81.97 (1.59)	88.77 (2.19)	76.20 (2.40)	**79.52** (3.80)	86.42 (4.57)	**73.64** (5.10)

class respectively have 115, 133 and 5471 images. For Skin198, we choose 70 most infrequent classes for this evaluation because there are less than 20 images in all these 70 classes. Table IV shows that our PCCT benefits the classification performance on infrequent classes in all three datasets, e.g., the mean F1 score is improved by 7.35% on the infrequent class of Skin7 compared with the baseline OCE. Similar improvements of the mean F1 score on infrequent classes are also observed on Skin198, ChestXray-COVID, and Kaggle EyePACs. Compared with the baseline OCE, our proposed PCCT achieves 7.46%, 9.9%, and 2.3% improvement on infrequent classes of Skin198, ChestXray-COVID, and Kaggle EyePACs respectively. Also, PCCT generally outperforms the newly-compared triplet-ratio loss by means of both overall performance and infrequent class classification performance. This can be probably contributed to the fact that PCCT is designed as a two-stage progressive framework which (a) designs a class-balanced triplet sampling strategy to deal with the problem that triplets whose anchors come from the infrequent class are rare; and (b) designs a class-center involved triplet strategy to deal with the problem that anchor and positive samples may be mapped far away. These strategies boost the representation of feature extractor in the coarse-to-fine triplet-based model training manner, which effectively alleviate the class imbalance issue.

C. Generalizability and Hyper-Parameter Evaluation

Our PCCT is experimentally proved to be robust to model architectures, as well as hyper-parameter variation such as output dimension and the margin α. Table V shows that when the CNN feature extractor varies in ResNet-50 [30], DenseNet-121 [33], Inception-v4 [34], VGG-19 [35], the proposed PCCT shows consistent performance improvement on the dataset Skin198. Tests with varying output dimensions (Fig. 2) also show that the proposed PCCT is still consistently better than corresponding

TABLE V
CLASSIFICATION PERFORMANCES OF PCCT ARE STABLY BETTER THEN THE COMPARED METHODS WHEN USING DIFFERENT MODEL ARCHITECTURES. COMPARISONS ARE PERFORMED ON SKIN198 DATASET

Methods	ResNet-50			DenseNet-121			Inception-v4			VGG-19		
	MF1	MCP	MCR	MF1	MCP	MCR	MF1	MCP	MCR	MF1	MCP	MCR
BCE [30] (2016)	51.91 (1.10)	56.41 (1.27)	52.12 (1.14)	62.91 (1.37)	65.95 (1.42)	64.00 (1.55)	50.22 (1.78)	53.73 (2.00)	50.83 (1.79)	51.16 (1.82)	53.81 (2.07)	52.51 (1.85)
WCE [5] (2019)	60.21 (1.36)	64.82 (1.34)	60.23 (1.12)	55.04 (1.74)	61.42 (2.01)	54.62 (1.68)	57.92 (1.71)	62.30 (1.24)	58.08 (2.05)	47.09 (7.91)	50.43 (7.71)	48.08 (7.67)
OCE [31] (2019)	59.77 (1.89)	64.87 (2.06)	59.34 (1.87)	57.72 (1.91)	63.86 (1.85)	56.96 (1.92)	56.79 (2.51)	61.83 (2.70)	56.49 (2.46)	50.85 (1.42)	53.99 (1.44)	52.09 (1.67)
WFCE [7] (2017)	53.28 (2.65)	58.31 (2.77)	53.34 (2.58)	43.03 (1.28)	46.70 (1.08)	44.00 (1.37)	49.88 (2.65)	53.53 (2.33)	50.44 (2.86)	37.13 (1.98)	39.68 (2.41)	38.61 (1.73)
TSD [10] (2020)	64.23 (1.54)	67.10 (1.90)	65.62 (1.56)	62.94 (1.54)	65.99 (1.47)	64.04 (1.71)	**59.49** (1.90)	62.13 (2.14)	**61.01** (2.06)	50.61 (1.93)	53.99 (1.88)	51.66 (2.06)
PCCT	**65.20** (1.49)	68.40 (1.36)	**66.02** (1.50)	62.62 (2.18)	64.94 (2.41)	64.26 (1.98)	59.25 (1.18)	62.43 (1.48)	60.12 (1.12)	**52.49** (2.33)	**55.39** (2.84)	**53.49** (2.50)
Efficient-PCCT	64.51 (1.65)	**68.76** (1.80)	64.74 (1.84)	**63.43** (2.01)	**66.70** (2.14)	**64.36** (1.98)	58.74 (1.43)	**63.12** (2.11)	60.03 (1.13)	42.63 (2.10)	46.09 (3.15)	44.38 (2.00)

TABLE VI
EXTENSION OF THE CLASS-CENTER BASED TRIPLET LOSS TO THE CLASS-CENTER BASED PAIR-WISE RANKING LOSS AND QUADRUPLET LOSS

Methods	Skin7			Skin198			ChestXray-COVID		
	MF1	MCP	MCR	MF1	MCP	MCR	MF1	MCP	MCR
Pair-wise (original)	84.54 (0.71)	88.95 (1.15)	81.14 (1.18)	53.87 (1.61)	60.37 (1.47)	52.59 (1.75)	81.03 (1.14)	87.48 (3.45)	76.70 (3.31)
Pair-wise (class centers)	85.81 (1.16)	88.70 (1.70)	83.53 (0.98)	64.82 (2.21)	67.51 (2.33)	66.06 (2.13)	90.05 (1.20)	92.89 (1.36)	87.72 (2.33)
Quadruplet (original)	84.42 (1.32)	89.03 (1.49)	81.15 (1.74)	63.08 (1.50)	67.59 (1.95)	63.23 (1.58)	82.51 (1.71)	90.36 (2.10)	77.19 (2.63)
Quadruplet (class centers)	86.11 (1.21)	88.79 (1.53)	84.08 (1.35)	65.00 (1.83)	67.77 (1.72)	66.11 (1.99)	90.36 (1.05)	92.02 (1.83)	88.99 (2.29)

Class-center based loss performs better than the corresponding original loss on all three datasets.

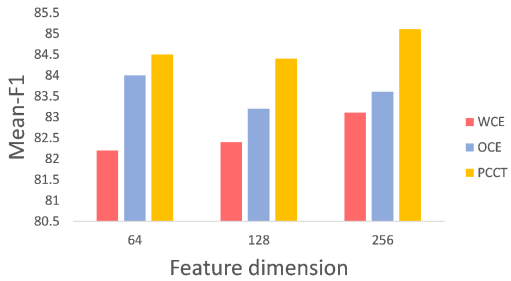


Fig. 2. Effect of output dimension on imbalanced classification. Skin7 is used here.

baseline methods with the same model architecture. Furthermore, when the margin α varies, the performance of the proposed PCCT is relatively stable on all three datasets (Fig. 3), e.g., when α alters from 0.1 to 0.9, the mean F1 score remains stable; its fluctuation is less than 1% on Skin7. These results are consistently supporting that the proposed two-stage PCCT method is stable and generalizable.

The role of class centers in the triplet loss is further demonstrated to be extensible to relevant metric learning, e.g., based on the pair-wise ranking loss and the quadruplet loss. As introduced in Section II-C, class centers can be easily used to replace the samples in the ranking loss and the quadruplet loss, resulting in the class-center based ranking loss and quadruplet loss respectively. As shown in Table VI, compared to the original pair-wise ranking loss and the quadruplet loss, class-center involved losses can help train better feature extractors on all three datasets. This suggests that class centers may be potentially applied in various metric learning strategies where multiple samples as training units are involved.

D. Ablation Study

Ablation studies were performed to evaluate the significance of the two stages, the sampling strategy, the triplet loss, and the trainable class centers. Experimental results demonstrates the significance of these designs.

The comparison in Table VII demonstrates the importance of the two-stage training strategy on three typical datasets (Skin7, Skin198, and ChestXray-COVID). For the effect of the first stage, performance gains on the three datasets (second VS sixth row in Table VII) consistently support that the pre-train procedure by the class-balanced triplet loss is crucial for the second-stage training by the class-center triplet loss. Compared with the only second-stage training (i.e., training the CNN feature extractor from scratch using only the class-center involved triplet loss), PCCT achieves higher performance with almost all three metrics (MF1, MCP, and MCR) on the three datasets with the help of the class-balanced triplet sampler. On the Skin 198

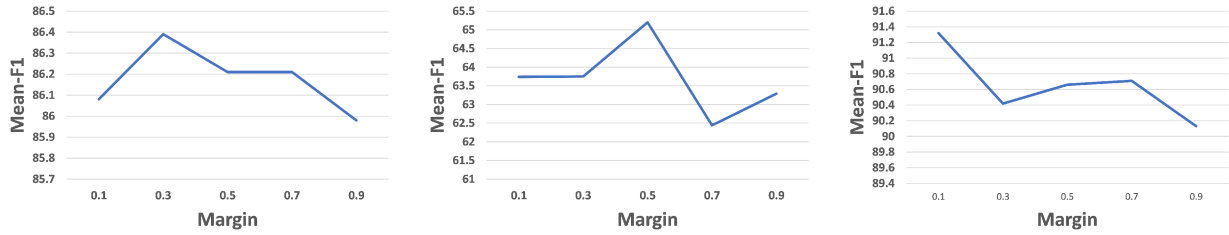


Fig. 3. Robustness of the proposed PCCT with respect to the margin α on all three datasets (from left: Skin7, Skin198, ChestXray-COVID).

TABLE VII
EFFECT OF THE TWO-STAGE TRAINING, THE SAMPLING STRATEGY, THE TRIPLET LOSS, AND THE LEARNABLE CLASS CENTERS IN PCCT

Methods	Skin7			Skin198			ChestXray-COVID		
	MF1	MCP	MCR	MF1	MCP	MCR	MF1	MCP	MCR
Only first-stage	84.02 (1.50)	88.96 (1.85)	80.44 (2.16)	64.25 (1.36)	68.55 (1.17)	64.39 (1.31)	81.59 (0.54)	87.43 (2.40)	77.37 (1.11)
Only second-stage	85.98 (0.96)	87.89 (2.02)	84.45 (0.40)	58.89 (1.34)	61.09 (1.54)	61.11 (1.17)	67.41 (4.97)	68.80 (5.90)	66.90 (4.68)
No balance sampling strategy	85.39 (0.74)	86.65 (1.07)	83.74 (0.78)	65.35 (1.30)	67.72 (1.10)	66.91 (1.32)	84.93 (1.63)	88.85 (2.80)	81.85 (1.50)
No triplet loss	84.10 (3.53)	84.21 (5.16)	82.46 (2.54)	53.40 (1.69)	56.31 (1.46)	52.60 (1.64)	86.79 (2.60)	91.34 (3.15)	83.06 (2.32)
No class centers	84.02 (1.50)	87.96 (1.85)	80.44 (2.16)	64.25 (1.36)	68.55 (1.17)	64.39 (1.31)	81.59 (0.54)	87.43 (2.40)	77.37 (1.11)
Two-stage (PCCT)	86.20 (1.07)	87.77 (1.54)	84.98 (0.75)	65.20 (1.49)	68.40 (1.36)	66.02 (1.50)	90.66 (1.43)	92.97 (1.58)	88.89 (2.70)

TABLE VIII
TRAINABLE CENTERS REQUIRE LESS COMPUTATIONAL TIME THAN COMPUTED CENTERS ON THREE DATASETS (SKIN7, SKIN198, AND CHESTXRAY-COVID)

Methods	Skin7	Skin198	ChestXray-COVID
Computed centers	60	47	205
trainable centers	45	38	154

The unit is second/epoch, i.e., The time consumption of training per epoch.

and ChestXray-COVID datasets, the performance improvement is more obvious. For example, on the Skin198 dataset, PCCT achieves a mean F1 score (MF1) of 65.20%, clearly higher than only using the second stage (which results in a MF1 of 58.89%). This demonstrates the class balancing strategy in triplet construction helps alleviate the imbalance issue. For the effect of the second stage, similar performance improvement has been observed (first v.s sixth row in Table VII), i.e., PCCT achieves higher performance with almost all three metrics on three datasets. For example, on the Skin7 dataset, PCCT achieves a mean F1 score (MF1) of 86.20%, clearly higher than only using the first stage (which results in a MF1 of 84.02%). This is probably because the class center-involved triplet strategy benefits triplet-based model training by avoiding anchor and positive samples to be mapped far away, which stabilizes the training when sample features show large variety in the frequent class.

To evaluate the significance of the class-balanced sampling strategy, an ablated version of the two-stage PCCT is carried out without balancing sampling strategy in the first stage, i.e., the first stage uses a random sampler to construct triplets. As shown in the third row of Table VII, the performance of this ablated version is lower than PCCT with all three metrics on the three datasets. This again shows the sampling strategy is beneficial to constructing class-balanced triplets and therefore benefits the classification performance.

For the ablation study on the triplet loss, we add an experiment by replacing the triplet loss with traditional cross-entropy loss in the two stages. As shown in Table VII (fourth row), the performance of the ablated PCCT significantly decreases, directly supporting the advantages of the proposed triplet loss in extracting discriminative feature representations and tackling the data imbalanced issue. The triplet loss directly helps samples from the same (or different) class to have similar (or dissimilar) features, which is beneficial for classification performance.

In addition, we conduct an experiment by replacing the trainable class center with traditional triplets. As shown in Table VII (fifth row), the classification performance is worse than that of the proposed PCCT, directly supporting the advantage of using class centers in triplet-based model training. Without using class centers as positives and negatives, the triplet quality may get worse because randomly selecting the positives and negatives in the triplets may result in too far anchor-positive distance, which damages the training stability.

E. Trainable Class Centers

We compare between the computed and the trainable class centers by means of classification performance and computational time. As shown in the last two rows of Table III and

TABLE IX
PERFORMANCE COMPARISON BETWEEN DIFFERENT INITIALIZATION METHODS ON THREE DATASETS

Methods	Skin7			Skin198			ChestXray-COVID		
	MF1	MCP	MCR	MF1	MCP	MCR	MF1	MCP	MCR
Zero	86.36 (0.95)	88.97 (1.51)	84.32 (0.70)	64.50 (1.65)	68.76 (1.80)	64.74 (1.84)	89.91 (1.44)	93.57 (2.42)	86.94 (2.60)
Gaussian	86.44 (0.64)	89.22 (0.77)	84.11 (0.57)	57.56 (5.59)	63.59 (4.60)	57.97 (5.96)	91.22 (1.40)	93.82 (2.69)	89.10 (1.63)
Uniform	86.24 (0.95)	89.28 (1.74)	83.86 (0.71)	64.32 (2.97)	68.43 (2.23)	64.64 (2.86)	90.55 (1.03)	93.46 (2.48)	88.22 (2.12)
First stage	**86.80** (0.96)	**89.72** (0.74)	**84.46** (1.32)	**65.68** (1.53)	**69.45** (2.22)	**66.11** (1.21)	**92.00** (1.15)	**94.10** (1.58)	**88.89** (2.70)

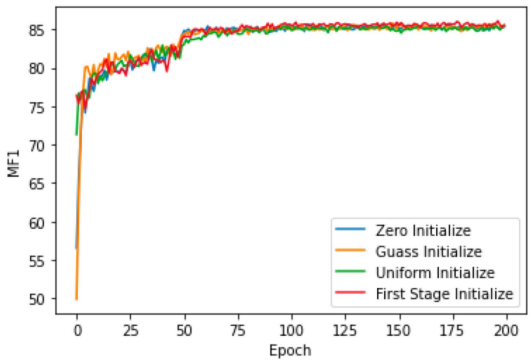


Fig. 4. The change in a representative evaluation metric (MF1) with training epochs showing the convergence speed is not affected by the initialization methods.

Table IV, the triplet loss with trainable class centers (Efficient-PCCT) performs similarly well compared to the original PCCT with computed class centers on all three datasets. However, the original PCCT needs to calculate the class centers at each training epoch, while the Efficient-PCCT simply updates the class center along with the model parameters, which is more efficient. As Table VIII shows, the training time decreases by approximately 1/3 with the trainable centers. Thus, considering its feasibility and effectiveness, trainable class centers may be a better choice than the estimated class centers computed at each training epoch especially when the training dataset is large.

We further explored the effect of initialization of the trainable centers. As shown in the first three rows of Table IX, zero initialization, Gaussian initialization, and uniform initialization for the class centers do not result in significant performance changes, i.e., the trainable centers are robust to these general initialization methods. However, as shown in the last row of Table IX, after training the first stage, if the centers of each class are calculated and used as the initialization of class centers for the second stage, the performance would be higher with all three metrics on all three datasets. This demonstrates that the first stage is beneficial to classification performance of the trainable centers. In addition, the convergence performances of the four initialization methods are tested. As shown in Fig. 4, the convergence speed is not affected by the initialization methods.

F. Discussions

From the above extensive evaluations, it is clear that the proposed two-stage learning framework with certain metric learning loss is effective in handling the class-imbalance issue and outperforms widely used strong baselines under various conditions. This is consistent with the previously reported two-stage learning strategy TSD for the class imbalance issue [10]. However, different from TSD which simply trains the model (mainly the feature extractor) with cross-entropy loss at the first stage and then applies certain class re-balancing strategy at the second stage, the class-balanced triplet loss and the class-center involved triplet loss in the proposed framework can further help train a better feature extractor by enforcing more compact within-class distribution and enlarging the separation between classes. Note that the class centers in the proposed framework can be learned together with model parameters, reducing the computational cost for class center estimate during model training. Consistent with our study, one recent work [36] used class center loss to help train a feature extractor for imbalanced image classification tasks. In addition, other types of metric learning losses could be applied as well in the two-stage learning framework, e.g., using contrastive loss [21] in the first stage, which will be part of our future exploration following this study.

The proposed framework focuses on training a class-balanced feature extractor. Therefore, it is complementary to many existing strategies which focus on the input side (e.g., re-sampling) or output side (re-weighting, focal loss, etc.) of the classifier. For example, our method may be combined with existing strategies to further enlarge separation between classes, e.g., with the help of the distribution-aware margin loss [6], or combined with data augmentation techniques like Mixup to further alleviate the data imbalance between classes [37]. Although promising performance has been achieved on various imbalanced medical datasets, the proposed two-stage framework with the triplet loss has an obvious limitation, i.e., relatively longer training time compared to the single-stage methods. Although the proposed Efficient-PCCT can decrease training time in the first stage, the second stage is still relatively time consuming. Replacing the triplet loss by other types of metric learning loss (e.g., contrastive loss or graph-based loss) could largely alleviate this issue. However, it is worth noting that the inference time is determined mainly based on the model backbone and therefore inference is in general near real-time for medical diagnosis.

IV. Conclusion

In this paper, we propose a two-stage method PCCT to handle the class imbalance issue. PCCT consists of two novel training stages. The triplet loss with the class-balanced triplet sampler is proposed to optimize the feature extractor model in the first stage, and then the class-center involved triplet loss is proposed to further fine-tune the feature extractor in the second stage such that the distribution of each class in the feature space becomes more compact and easily separated from each other. The classification performance on imbalanced datasets, stability, and generality of the proposed PCCT on various model backbones, output sizes, and hyperparameter settings are demonstrated by extensive experiments. The class-center idea has also been easily extended to other relevant metric learning approaches. We expect that this two-stage method will help effectively develop intelligent diagnosis systems for both common and rare diseases.

References

[1] P. Tschandl, C. Rosendahl, and H. Kittler, "The HAM10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions," *Sci. Data*, vol. 5, 2018, Art. no. 180161.

[2] S. Shilaskar and A. Ghatol, "Diagnosis system for imbalanced multi-minority medical dataset," *Soft Comput.*, vol. 23, pp. 4789–4799, 2019.

[3] A. Jain, S. Ratnoo, and D. Kumar, "Addressing class imbalance problem in medical diagnosis: A genetic algorithm approach," in *Proc. Int. Conf. Inf., Commun., Instrum. Control*, 2017, pp. 1–8.

[4] M. Buda, A. Maki, and M. Mazurowski, "A systematic study of the class imbalance problem in convolutional neural networks," *Neural Netw.*, vol. 106, pp. 249–259, 2018.

[5] Y. Cui, M. Jia, T.-Y. Lin, Y. Song, and S. Belongie, "Class-balanced loss based on effective number of samples," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 9268–9277.

[6] K. Cao, C. Wei, A. Gaidon, N. Arechiga, and T. Ma, "Learning imbalanced datasets with label-distribution-aware margin loss," in *Proc. 33rd Int. Conf. Neural Inf. Process. Syst.*, 2019, pp. 1567–1578.

[7] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2980–2988.

[8] Z. Liu, Z. Miao, X. Zhan, J. Wang, B. Gong, and S. Yu, "Large-scale long-tailed recognition in an open world," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 2532–2541.

[9] Y. Yang and Z. Xu, "Rethinking the value of labels for improving class-imbalanced learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 19290–19301.

[10] B. Kang et al., "Decoupling representation and classifier for long-tailed recognition," in *Proc. Int. Conf. Learn. Representations*, 2020, pp. 1–16.

[11] B. Zhou, Q. Cui, X.-S. Wei, and Z. Chen, "BBN: Bilateral-branch network with cumulative learning for long-tailed visual recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 9716–9725.

[12] Y. Zhang, X.-S. Wei, B. Zhou, and J. Wu, "Bag of tricks for long-tailed visual recognition with deep convolutional neural networks," in *Proc. 35th AAAI Conf. Artif. Intell.*, 2021, pp. 3447–3455.

[13] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 815–823.

[14] A. Hermans, L. Beyer, and B. Leibe, "In defense of the triplet loss for person re-identification," 2017, *arXiv:1703.07737*.

[15] C. Song, Y. Huang, W. Ouyang, and L. Wang, "Mask-guided contrastive attention model for person re-identification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 1179–1188.

[16] X. He, Y. Zhou, Z. Zhou, S. Bai, and X. Bai, "Triplet-center loss for multi-view 3D object retrieval," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 1945–1954.

[17] W. Ge, W. Huang, D. Dong, and M. Scott, "Deep metric learning with hierarchical triplet loss," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 269–285.

[18] P. Laiz, J. Vitrià, H. Wenzek, C. Malagelada, F. Azpiroz, and S. Seguí, "WCE polyp detection with triplet based embeddings," *Computerized Med. Imag. Graph.*, vol. 86, 2020, Art. no. 101794.

[19] T. Y. Liu and J. Feng, "Triplet contrastive learning for brain tumor classification," 2021, *arXiv:2108.03611*.

[20] W. Lei, R. Zhang, Y. Yang, R. Wang, and W.-S. Zheng, "Class-center involved triplet loss for skin disease classification on imbalanced data," in *Proc. Int. Symp. Biomed. Imag.*, 2020, pp. 1–5.

[21] S. Chopra, R. Hadsell, and Y. LeCun, "Learning a similarity metric discriminatively, with application to face verification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2005, vol. 1, pp. 539–546.

[22] W. Chen, X. Chen, J. Zhang, and K. Huang, "Beyond triplet loss: A deep quadruplet network for person re-identification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 403–412.

[23] S. Ding, L. Lin, G. Wang, and H. Chao, "Deep feature learning with relative distance comparison for person re-identification," *Pattern Recognit.*, vol. 48, no. 10, pp. 2993–3003, 2015.

[24] M. Hayat, S. Khan, S. W. Zamir, J. Shen, and L. Shao, "Gaussian affinity for max-margin class imbalanced learning," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 6468–6478.

[25] A. Filos et al., "A systematic comparison of Bayesian deep learning robustness in diabetic retinopathy tasks," 2019, *arXiv:1912.10481*.

[26] N. Codella et al., "Skin lesion analysis toward melanoma detection 2018: A challenge hosted by the international skin imaging collaboration (ISIC)," 2019, *arXiv:1902.03368*.

[27] X. Sun, J. Yang, M. Sun, and K. Wang, "A benchmark for automatic visual classification of clinical skin disease images," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 206–222.

[28] J. P. Cohen, P. Morrison, L. Dao, K. Roth, T. Q. Duong, and M. Ghassemi, "COVID-19 image data collection: Prospective predictions are the future," 2020, *arXiv:2006.11988*.

[29] Kaggle, "Diabetic retinopathy detection challenge," 2015. [Online]. Available: https://www.kaggle.com/c/diabetic-retinopathy-detection

[30] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.

[31] J. Byrd and Z. Lipton, "What is the effect of importance weighting in deep learning?," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 872–881.

[32] S. Hu, K. Wang, J. Cheng, H. Tan, and J. Pang, "Triplet ratio loss for robust person re-identification," in *Proc. Pattern Recognit. Comput. Vis.*, 2022, pp. 42–54.

[33] G. Huang, Z. Liu, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2261–2269.

[34] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," in *Proc. 31st AAAI Conf. Artif. Intell.*, 2017, pp. 4278–4284.

[35] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. Learn. Representations*, 2015, pp. 1–14.

[36] J. Cui, Z. Zhong, S. Liu, B. Yu, and J. Jia, "Parametric contrastive learning," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2021, pp. 715–724.

[37] Z. Zhong, J. Cui, S. Liu, and J. Jia, "Improving calibration for long-tailed recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 16489–16498.