Similar Classes Latent Distribution modelling-based Oversampling Method for Imbalanced Image Classification

Wei Ye
Guilin University of Technology
Minggang Dong (d2015mg@qq.com)
Guilin University of Technology
Yan Wang
Guilin University of Technology
Guojun Gan
Guilin University of Technology
Deao Liu
Guilin University of Technology

Research Article

Keywords: Imbalanced classification, Oversampling, latent distribution, Similar classes, Boundary samples

Posted Date: August 24th, 2022

DOI: https://doi.org/10.21203/rs.3.rs-1977513/v1

Similar Classes Latent Distribution modelling-based Oversampling Method for Imbalanced Image Classification

Wei Ye[1,2], Minggang Dong[1,2*], Yan Wang[1,2], Guojun Gan[1,2] and Deao Liu[1,2]

[1*]School of information science and engineering, Guilin University of Technology, Guilin, 541004, China.
[2] Guangxi Key Laboratory of embedded technology and intelligent system, Guilin, 541004, China.

*Corresponding author(s). E-mail(s): d2015mg@qq.com;

Abstract

Learning an unbiased classifier from imbalanced image datasets is a challenging task, since the classifier may strongly bias towards majority classes. To address this issue, some deep generative models-based oversampling methods have been proposed. However, most methods pay little attention to the decision boundary, which may contribute tiny to learning an unbiased classifier. In this paper, we focus on the decision boundary and propose a similar classes latent distribution modelling-based oversampling method. Specifically, first, we model each class as different von Mises-Fisher distributions, thereby aligning feature learning with the class distributions. Furthermore, we develop a distance minimization loss function, which makes similar classes closer in latent space. The generator can learn more shared latent features from the decision region. In addition, we propose a boundary sampling strategy, which uses latent variables between similar classes to generate boundary samples for data balancing. Experiments on four imbalanced image datasets show that the proposed method achieves promising performance in terms of Recall, Precision, F1-score and G-mean.

Keywords: Imbalanced classification, Oversampling, latent distribution, Similar classes, Boundary samples

1 Introduction

Image classification has been an attractive research field of computer vision in recent years [1]. The improvement of image classification performance relies on large-scale datasets with a relatively balanced class distribution, such as ILSVRC 2012 [2] and MS COCO [3]. However, in the real world, image datasets are often imbalanced [4], in which a few classes (majority class) have the majority of samples while others (minority class) are scarce [5]. When using imbalanced image datasets to train a classifier, the traditional model may be skewed toward learning the features of the majority class [6, 7], resulting in poor classification performance for minority classes.This is called the imbalanced learning problem. In practice, imbalanced image datasets are commonly encountered in anomaly detection [8, 9], medical image classification [10, 11], and object detection [12]. As a result, it is a significant challenge that both industry and academia must face [7].

Solving the imbalanced learning problem aims to train an unbiased classifier that accurately predicts the class labels of data samples [13]. Researchers have proposed several methods to handle this problem. Among these approaches, data-level oversampling is regarded as one of the most effective methods [14, 15]. It balances the dataset by increasing minority class samples, reducing the impact of the imbalanced distribution on the classifier. The most common oversampling method is Synthetic Minority Over-sampling Technique (SMOTE) [16, 17]. Following that, more SMOTE-based oversampling methods [18–21] have been proposed, which attempt to identify the boundary between the minority and majority classes to generate more representative samples. However, traditional oversampling methods use Euclidean distance as a similarity measure, so they are unsuitable for handling high-dimensional imbalanced datasets such as images and audio [17].

Recent advances in deep generative models, particularly generative adversarial networks (GAN) [22] and variational autoencoders (VAE) [23], have brought new opportunities for imbalanced learning. Some GAN-based and VAE-based models have been proposed to generate synthetic image samples in the minority class [13, 24–29]. However, most of these works do not consider concentrating the generated samples in decision regions where features are difficult to classify, resulting in the generated samples contributing tiny to training an unbiased classifier [30]. Recent research demonstrates that samples near the decision boundary (called boundary samples in this paper) are more critical for training the classifier than those far from the decision boundary [18, 31]. GAMO [30] and DVAAN [32] are proposed to generate boundary minority class samples. However, GAMO may suffer from mode collapse, resulting in a lack of diversity in the generated samples, which may not increase the classifier performance significantly [33]. DVVAN is highly dependent on the selection of appropriate similar classes. If two inappropriate classes are selected, it will generate low-quality samples. Besides, DVVAN is a binary framework, which is not suitable for the multi-class imbalanced image dataset.

Hence, this article proposes a similar classes latent distribution modelling-based oversampling method, which generates more boundary minority class samples to help train an unbiased classifier. Specifically, we design a similar classes modelling network (SCN), which is an improved network of VAE-GAN [34] and consists of two steps. In the first step, the von Mises-Fisher (vMF) distribution is introduced as prior and variational posterior distribution of SCN, which can prevent the KL divergence from forcing all latent variables to concentrate on one point. Therefore, the encoder can model each class as different vMF distributions, which aligns the feature learning with the class distributions to distinguish the minority class from the majority class, effectively improves the quality of generated samples [32, 35]. In the second step, a Distance Minimization loss function (DM loss) is proposed to reduce the inter-class distance of similar classes, making their latent distributions closer. The generator learns about shared latent features from the decision region of similar classes. For oversampling, we propose a boundary sampling strategy, which forms new sampling regions in the middle of two similar classes. After training convergence, the generator generates boundary samples by using the latent variables in this region.

Our main contributions can be summarized as follows:

1. We propose a similar class latent distribution modeling network (SCN), which models each class as a different latent distribution to clearly distinguish the minority class from the majority class.
2. We design a Distance Minimization loss function (DM loss) for the encoder, which makes similar classes closer in the latent space. Consequently, the generator can learn shared latent features from their decision region.
3. A boundary sampling strategy is proposed to generates boundary samples by using the latent variables in the middle of two similar classes.
4. Extensive experiments on four imbalanced image datasets demonstrate the superiority of the proposed method.

The rest of this paper is structured as follows: Section 2 briefly discuss related work. Section 3 describes the proposed method in detail. In Section 4, the proposed method is evaluated by comparison and ablation experiments. Section 5 concludes this article.

2 Related Work

Over the last few decades, experts and scholars have proposed various solutions to the imbalanced learning problem. These approaches are divided into two categories: data-level and algorithm-level. First, some algorithm-level and dataset-level methods are introduced, and then recent research on boundary samples-based oversampling methods is presented. Moreover, some concepts used in this paper are also introduced.

2.1 Algorithm-level approach

The algorithm-level approach aims to modify existing learning algorithms to reduce bias toward the majority class. Cost-sensitive learning [6], which considers the relationship between class-wise costs and misclassified samples, thereby increasing the sensitivity of the classifier to the minority class [36]. When misclassifying minority samples, a higher cost can be set by the cost matrix [7]. Cost-sensitive learning representative works are as follows. Khan et al. [5] jointly optimized the misclassification cost and network weight parameters. The focal loss proposed by Lin et al. [12] makes the network pay more attention to misclassified samples by weighing the classification loss of different classes. Reference [37] propose a class rectification loss in conjunction with hard sample mining, aiming to identify the sampling boundaries of the minority class, thereby reducing the dominant effect of the majority class. Although cost-sensitive learning can readily be applied to deep learning, determining the value of the cost matrix is difficult [7].

2.2 Data-level approach

The data-level approach rebalances the dataset distribution by increasing minority class instances or decreasing majority class instances. Deep generative models have been used successfully to augment datasets in recent years [38]. Douzas et al. [25] propose using CGAN to generate samples with specified labels, whereas Odena et al. [39] propose using auxiliary classifiers to improve the quality of generated samples. However, these methods may generate the wrong class samples in extreme cases (high imbalance rate). Ali-Gombe et al. [28] propose MFC-GAN, which modifies the training objectives of ACGAN [39] by adding real and fake class labels, forcing the generator to generate the correct class samples. However, the classifier and discriminator in MFC-GAN share the same network. Suh et al. [17] believe that the shared network will cause instability in GAN training and reduce the quality of generated samples. As a result, they treat the classifier as an independent network structure in their proposed CEGAN and train the backbone network with WGAN-GP. [40]. Furthermore, the researchers propose combining GAN with VAE to reduce the difficulty of mapping from simple random distribution to complex data distribution. This strategy was used to generate samples by Balance Gan (Bagan) [41] and Data Augmentation Generative Adversarial Networks (DA-GAN) [26].

2.3 Boundary samples-based oversampling approach

According to research, samples located near the decision boundary are more likely to be misclassified, so learning a robust classification model requires boundary samples [31]. For this reason, this paper focuses on the study of generating boundary samples. Some traditional methods of synthesizing boundary samples have been proposed previously, these methods include Borderline-smote [18], Safe-level-smote [19] , and ADASYN [21]. In deep learning, Mullick

et al. [30] and Guo et al. [32] proposed GAMO and DVVAN to generate boundary samples, respectively. GAMO consists of a generator, discriminator and classifier. The generator and classifier play an adversarial game such that the generator generates samples that lie near the decision boundary. These samples help the classifier learn boundary information. Although their proposed models perform well on imbalanced image datasets, recent studies suggest that GAMO may suffer from the mode collapse [33].

Compared with GAMO, the DVVAN provides a different idea.First, artificially select two similar classes and use the encoder to model the two classes as latent distributions with opposite means, then generate boundary samples using boundary latent variables. Although this method can avoid mode collapse to a certain extent, they do not consider the subjective bias caused by the artificial selection of similar classes, which makes the model generate low-quality samples. In addition, DVVAN is a binary framework, which is unsuitable for multi-class imbalanced datasets.

2.4 VAE-GAN

Our proposed network is an improved VAE-GAN [34] , it is necessary to introduce related concepts before delving into the proposed method.

VAE-GAN combines variational autoencoders and generative adversarial networks. It uses GAN to generate high-quality samples, and VAE models the data into a latent space. Therefore, the objective function of VAE-GAN consists of the adversarial loss of GAN and the negative evidence lower bound (ELBO) of VAE.

$$L_{ELBO} = \mathbb{E}_{z\sim q_\phi(z|x)}\left[\|F_D(x) - F_D(G(x))\|_2^2\right] + KL\left[q_\phi(z \mid x)\|p(z)\right] \tag{1}$$

$$\begin{aligned} L_{GAN} = \mathbb{E}_{x\sim p_r(x)} \log D(x) + \mathbb{E}_{z\sim q_\phi(z|x)} \log(1 - D(G(z))) \\ + \mathbb{E}_{z\sim p(z)} \log(1 - D(G(z))) \end{aligned} \tag{2}$$

The first term of Eq. (1) is the feature-wise reconstruction loss, which reconstructs z into data samples by using the features extracted from the ℓth layer of discriminator. $F_D(\cdot)$ denotes the features extracted from the ℓth layer of discriminator.The second term is the KL divergence, which forces the approximate posterior distribution $q_\phi(z \mid x)$ to match $p(z)$, $p(z)$ is usually assumed to be $\mathcal{N}(0, \mathrm{I})$. Furthermore, as shown in Eq. (2), VAE-GAN needs to sample from the data distribution $p_r(x)$, the posterior distribution $q_\phi(z \mid x)$, and the prior distribution $p(z)$, respectively, during adversarial training.

3 Proposed method

In this section, we propose a similar classes latent distribution modeling-based oversampling method. The innovation of our method is to generate boundary samples by using the latent variables, which draw from the decision region of similar classes. To this end, subsection 3.1 introduce SCN how to model

a latent space of similar classes. Subsection 3.2 explain how to sample latent variables from decision regions. Subsection 3.3 gives an example to illustrate the proposed method better.

3.1 Overall network

As shown in Fig.1, SCN is an improved VAE-GN network, including five parts: 1) the encoder network E; 2) the generator network G; 3) the discriminator network D; 4) Classifier network C; 5) the similar classes selector C_{sim}.

We feed image samples, class labels, and similar classes labels into the encoder network E, which models samples of different classes as different latent distributions. The generator network G learns to generate class-specific samples from different latent distributions. The discriminator network D helps G improve the quality of the generated samples through adversarial training. The Classifier C ensures that G generates samples of the correct class. Furthermore, for the generator G to learn about shared latent features of similar classes, the encoder E gradually guides the latent distribution of similar classes closer during the training process. By exploiting the latent variables of the decision region, the generator can generate boundary sample.

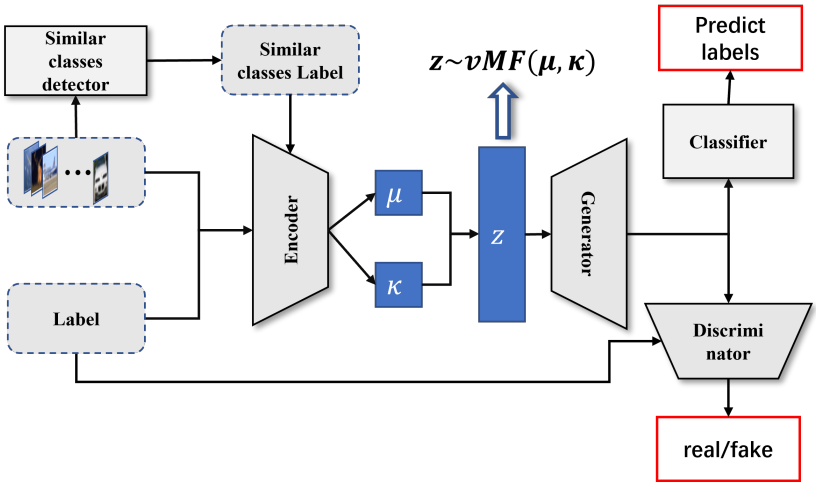


Fig. 1 The overall network architecture of SCN.

Overall, as shown in Fig.2, training an SCN can be viewed as two processes: 1) modelling each class as a different latent distribution (details in subsection 3.1.1); 2) guiding the latent distributions of similar classes closer (details in subsection 3.1.2); The training details and objective function of SCN are introduced in subsection 3.1.3.

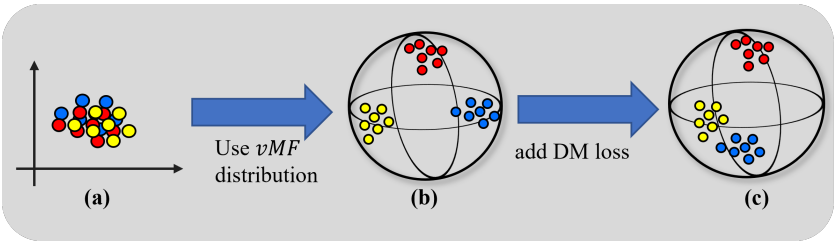


Fig. 2 (a) KL divergence concentrates all latent variables at one point. (b) After using the vMF distribution, each class is represented as a different distribution in the hypersphere space. (c) DM loss guides the latent distribution of similar classes closer.

3.1.1 vMF distribution-based latent variable modelling

Our goal is to model each class as a latent distribution with clear boundaries, thereby aligning feature learning with the class distribution. Specifically, consider a dataset $\mathcal{D} = \left\{(x_i, y_i) \mid x_i \in \mathbf{R}^d, y_i \in \{1, 2, ..., C\}, i = 1, \dots n\right\}$, x_i is a training sample and y_i is the corresponding class label. For any $(x_i, y_i = c)$, we force,

$$\begin{cases} z_i^c = \mathrm{Enc}(x_i \mid y_i = c) \sim q_\phi(z \mid x_i, y_i = c) \\ \tilde{x}_i^c = \mathrm{Dec}(z_i^c) \sim p_\theta(x \mid z_i^c) \end{cases} \tag{3}$$

where $\mathrm{Enc}(\cdot)$ and $\mathrm{Dec}(\cdot)$ represent the encoder and decoder(also the generator of GAN), respectively. $q_\phi(\cdot)$ and $p_\theta(\cdot)$ are parameterized by the encoder and decoder, respectively.

However, the original VAE-GAN does not meet the above requirements. Eq. (4) describes the value range of each part of ELBO, where the reconstruction loss and KL divergence are both non-negative.

$$\min \underbrace{\mathbb{E}_{z\sim q_\phi(z|x)}\left[\|F_D(x) - F_D(G(x))\|_2^2\right]}_{\textbf{non-negative}} + \underbrace{KL\left[q_\phi(z \mid x) \mid p(z)\right]}_{\begin{cases} KL\left[q_\phi(z \mid x)p(z)\right] > 0, q_\phi(z \mid x) \neq p(z) \\ KL\left[q_\phi(z \mid x)p(z)\right] = 0, q_\phi(z \mid x) = p(z) \end{cases}} \tag{4}$$

When training VAE-GAN, the KL divergence is continuously forces $q_\phi(z \mid x)$ to match $p(z)$. The prior distribution usually uses $\mathcal{N}(0, I)$, which causes the encoder to concentrate the latent distribution of all classes on one point, forming a cluster that cannot distinguish the class boundary (see Fig.2(a)). To address this issue, it is necessary to introduce a prior distribution that should not affect the mean of the posterior when optimizing the KL divergence.

The von Mises-Fisher (vMF) distribution is also known as the standard Gaussian distribution defined on a $d-1$ dimensional hypersphere with a sample space of $S^{d-1} = \left\{z \mid z \in R^d, \|z\| = 1\right\}$. The probability density is defined as follows:

$$q(z \mid \mu, k) = C_d(k) \exp\left(k\mu^T z\right), C_d(k) = \frac{k^{d/2-1}}{(2\pi)^{d/2} \mathrm{I}_{d/2-1}(k)} \tag{5}$$

here μ denotes the mean direction and $k \in \mathbb{R}_{\geq 0}$ denotes the concentration of the latent variable near μ. I_d is a modified Bessel function of the first kind of order d, and $C_d(k)$ is a normalizing constant. When $k = 0$, it represents the uniform distribution on the hypersphere.

SCN introduces $vMF(\cdot, k = 0)$ as prior distribution, and $vMF(\mu, k)$ as posterior distribution to depict the latent representation. By using the derivation method of [42] to obtain the following KL divergence.

$$KL[vMF(\mu, k)\|vMF(\cdot, 0)] = k\frac{I_{d/2}(k)}{I_{d/2-1}(k)} + \log C_d(k) - \log\left(\frac{2\left(\pi^{d/2}\right)}{\tau(d/2)}\right)^{-1} \quad (6)$$

Crucially, the KL divergence term in Eq. (6) only depends on k. μ is only optimized in reconstruction loss. The encoder can effectively learn to model different latent distributions for each class based on class information without being affected by KL divergence (see Fig.2(b)).

3.1.2 The distance minimization loss function

If the class distributions are far apart, it is difficult for generator to learn the shared latent features of similar classes, making it difficult to generate boundary samples. In this section, a Distance Minimization loss function (DM loss) is proposed to guide the latent distribution of similar classes closer.

Specifically, first, SCN uses the similar classes selector C_{sim} to obtain the similar classes labels of the training samples. C_{sim} is a classification model pretrained with imbalanced dataset. When a sample $(x, y = c)$ inputs to C_{sim}, it outputs the similar classes label y_{sim} of x based on the highest misclassification probability. This method of selecting similar classes can reflect that the classifier has extracted some shared features and the skewed direction of the decision boundary. Furthermore, according to the conclusions provided by [23] and [31] on the relationship between latent variables and generated samples, it is easy to infer the following corollaries:

Corollary 1. The distance between two latent variables is proportional to the distance between the corresponding samples

$$distance(z_1, z_2) \propto distance(x_1, x_2) \quad (7)$$

This proportional relationship is described in Fig.3.

Corollary 2. When the generated sample $\tilde{x}_y$ is distributed near its similar class in the feature space, the classification probability of $\tilde{x}_y$ satisfies:

$$p_y = \frac{\exp\left(w_y^T \tilde{x}_y\right)}{\sum_1^c w_i^T \tilde{x}_y} \approx p_{sim} = \frac{\exp\left(w_{sim}^T \tilde{x}_y\right)}{\sum_1^c w_i^T \tilde{x}_y} \quad (8)$$

Where w_y and w_{sim} represent the classification weight vector of the current class y and its similar class y_{sim} respectively. According to [43], the

weight vector w is normalized to 1, and the posterior classification probability becomes:

$$p_y = \|\tilde{x}_y\| \cos(\mathcal{A}_y) \approx p_{sim} = \|\tilde{x}_{sim}\| \cos(\mathcal{A}_{sim}) \tag{9}$$

Where $\mathcal{A}_y$ represents the angle between $\tilde{x}_y$ and w_y,$\mathcal{A}_{sim}$ represents the angle between $\tilde{x}_y$ and w_{sim}. The classification result only depends on $\mathcal{A}_y$ and $\mathcal{A}_{sim}$. If $\mathcal{A}_y \approx \mathcal{A}_{sim}$, it means that the generated sample $\tilde{x}_y$ is close to the decision boundary. In addition, the angle $\mathcal{A}_{y,sim}$ between w_y and w_{sim} can be used to approximate the distance between class y and class y_{sim} in the feature space (see Fig.3). Therefore, if the angle $\mathcal{A}_{y,sim}$ is smaller, the similar classes are closer in the feature space.

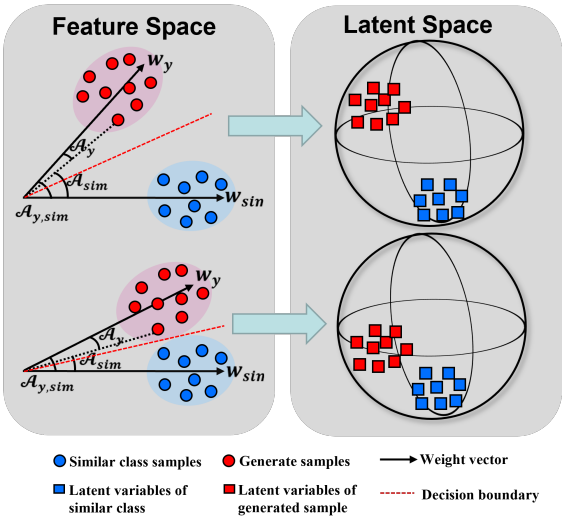


Fig. 3 When the generated sample and its similar class are far from the decision boundary, their latent distributions are also far from it. Conversely, when both the generated sample and its similar class are concentrated near the decision boundary, their latent distributions are also concentrated near it.

Based on the above two corollaries, the encoder is trained to minimize $\mathcal{A}_{y,sim}$. The encoder makes the latent distributions of similar classes closer, allowing the generator to learn more simlar feature from latent variables in the decision region, This means that the generator can map the latent boundary variables to boundary samples by adversarial training. Furthermore, If the angle $\mathcal{A}_{y,sim}$ is too small, the latent distributions of similar classes may overlap. In this case, it is difficult for the discriminator to separate the different class features from the overlapping parts, which causes the generator to generate some samples of the wrong class Therefore, we prevent the latent distributions from being too close by restricting $\mathcal{A}_y \leq \mathcal{A}_{\text{sim}}$, thus making sure that the generator generates samples of the correct class. The DM loss can be described

as follows:

$$\begin{aligned} &\min \mathcal{A}_{y,\text{sim}} \\ &\text{s.t } \cos(\mathcal{A}_y) - \cos(\mathcal{A}_{sim}) \geq \delta \end{aligned} \tag{10}$$

The margin parameter δ is used to ensure that the generated samples $\tilde{x}_y$ can be correctly classified by the classifier. In experiments, δ is set to 0.2 can effectively ensure that the class boundary is distinguishable. The Eq. (10) can be converted into the following form:

$$L_{DM} = \mathcal{A}_{y,sim} + \max(0, \cos(\mathcal{A}_y) - cos(\mathcal{A}_{sim}) - \delta) \tag{11}$$

The DM loss is added to the objective function of the encoder, when the latent distributions overlap, making the generator generate samples of the wrong class, the second term of L_{DM} acts like a penalty term. It forces the encoder to pay more attention to the class boundary and then gradually separates the overlapping parts of the latent distributions. The role of DM loss is shown in Fig.2(c).

3.1.3 Training of SCN

According to the conclusion of Suh et al. [17], the discriminator and the classifier are designed as two independent networks to stabilize the network training. In addition, a features-wise reconstruction loss is used, which enables the reconstructed samples have more details.The objective function of SCN is defined as follows:

$$\begin{aligned} L_D = &\mathbb{E}_{x\sim p_r(x)}[\log D(x \mid y)] + \mathbb{E}_{z\sim q_\phi(z|x)}[\log(1 - D(G(z \mid y)))] \\ &+ \mathbb{E}_{z\sim p(z)}[\log(1 - D(G(z \mid y)))] \end{aligned} \tag{12}$$

$$L_G = \mathbb{E}_{z\sim q_\phi(z|x)}[\log D(G(z))] + \mathbb{E}_{z\sim q_\phi(z|x)}\left[\|F_D(X) - F_D(G(z))\|_2^2\right] \tag{13}$$

$$\begin{aligned} L_E = &\mathbb{E}_{z\sim q_\phi(z|x,y=c)}\left[\|F_D(x \mid y) - F_D(G(z) \mid y)\|_2^2\right] \\ &+ KL[vMF(\mu, k)\|vMF(\cdot, 0)] \end{aligned} \tag{14}$$

$$L_C = \mathbb{E}_{x\sim p_r(x)}[\log C(y \mid x)] + \mathbb{E}_{z\sim q_\phi(z|x)}[\log C(y \mid G(z))] \tag{15}$$

Note that L_D,L_G,L_E,L_c are the loss functions of discriminator, generator, encoder, and classifier, respectively. Since there is a lack of minority classes in the real samples, only training the classifier with these samples will easily overfit the majority class. Therefore, our classifier is trained with real samples and generated samples. The training details of SCN are summarized in Alg 1. To stably model different distributions for each class on the hypersphere, DM loss should be added to the encoder every 5 epochs to gradually make similar classes closer.

Algorithm 1 training of SCN. default values $n_{GD} = 5$

Input: X: a set of training samples, m: the batch size
Output: $\tilde{x}$: generated samples

Initializing θ_D,θ_E,θ_G,θ_C
for $t = 1$ to n **do**
  if t mod $n_{GD} == 0$ **then**
    Sample $\{x^i\}_{i=1}^m \sim p_r$ a batch from X,and labels $\{y^i\}_{i=1}^m$.
    $\tilde{z} \leftarrow E(x \mid y)$
    $\tilde{x} \leftarrow G(z)$
    $y_{sim} \leftarrow C_{\text{sim}}\left(\tilde{\mathrm{x}}^i\right)$
    $\theta_G \leftarrow \text{Adam}\left(\nabla_{\theta_G} L_G\right)$ update the decoder
    $\theta_E \leftarrow \text{Adam}\left(\nabla_{\theta_E}(L_E + L_{DM})\right)$ update the encoder
  else
    Smaple $\{x^i\}_{i=1}^m \sim p_r$ a batch from X and labels $\{y^i\}_{i=1}^m$.
    Sample $\{z^i\}_{i=1}^m \sim p_z$ a generated batch.
    $\tilde{z} \leftarrow E(x \mid y)$
    $\tilde{x} \leftarrow G(\tilde{\mathrm{z}})$
    $\bar{x} \leftarrow G(z)$
    $\theta_D \leftarrow \text{Adam}\left(\nabla_{\theta_D} L_D\right)$ update the discriminator
    $\theta_C \leftarrow \text{Adam}\left(\nabla_{\theta_C} L_C\right)$ update the classifier
    $\theta_G \leftarrow \text{Adam}\left(\nabla_{\theta_G}(L_G + L_C)\right)$ update the generator
    $\theta_E \leftarrow \text{Adam}\left(\nabla_{\theta_E} L_E\right)$ update the encoder
  end if
end for

3.2 Boundary sampling strategy

When the proposed network is trained to converge, the maximum likelihood estimation was applied to obtain the distribution center for each class. The maximum likelihood optimization problem can be written as follow:

$$\max_{\mu} \mathcal{L}(\mu) = N^i k \mu^T \bar{z}^i + N^i \log C_d(k) \tag{16}$$

Where $\bar{z}^i = \frac{1}{N}\sum_{j=1}^{N^i} z_j^i$, z^i represents the latent variable of the i-th class,and N^i represents the number of z^i. By derivation of Eq. (16), the distribution center is $\mu^* = \frac{\bar{z}^i}{\|\bar{z}^i\|}$.

Since the latent distribution is modeled on the unit hypersphere, the consine distance can be used as a distance measure. Each time choose a minority class center μ_{near} as the "anchor" to find μ_{near}, which is the center with the shortest cosine distance from μ_{near}, then the new sampling center is $\mu_{new} = \alpha \frac{\mu_{\min} - \mu_{near}}{2}$. The sampling distribution formed by μ_{new} is located in the decision region of two similar classes, so the generator can generate boundary minority class samples by using the latent variables from this region. These

samples are closer to decision boundary in feature space. The detailed process of boundary sampling strategy is summarized in Alg 2.

Algorithm 2 boundary sampling strategy.

Input: $\{\mu^i\}_{i=1}^n$: latent distribution centres, m: number of minority classes
Output: boundary latent variables z

for $j = 1$ to m **do**
  $distance = 2$
  for $i = 1$ to n **do**
    $temp = 1 - cos(\mu_{min}^j, \mu^{i,i\neq j})$
    if $temp < distance$ **then**
      $distance = temp$
      $\mu_{near} = \mu^{i,i\neq j}$
    else
      continue
    end if
  end for
  $\mu_{new} = \alpha \frac{\mu_{\min} - \mu_{near}}{2}$
  sample z^j from $vMF(\mu_{new}, k)$
end for

In addition, the parameter $\alpha \in (0, 1]$ can be used to adjust the position of the sampling center, thereby controlling the style of the generated image. As shown in Fig.4, the digital "9" and the digital "4" are two similar classes. When the SCN generates digital "9", α can control how similar the digital "9" is to the digtal "4". Incorporating these boundary samples with similar features into the training set can help the classifier learn a more robust decision boundary. In experiments, the value of α is to 0.6 can achieve better results.

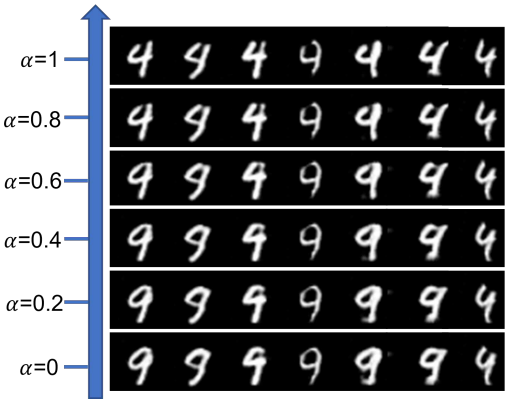


Fig. 4 An example of the MNIST dataset: digital "9" and digital "4" are similar classes, in which the digital "9" is the generated sample. When the value of α is larger, the sampling region is closer to digital "4", so the generated sample is more like the digital "4".

3.3 An example of the proposed method

This section assumes a simple example to illustrate the proposed method better. As shown in Fig.5, given an imbalanced image dataset, where the digital "6" and "7" are the majority class, the digital "5" is the minority class. The digital "5" and "6" are similar classes.The circles represent latent variables, and the blue dotted line represents the decision boundary of imbalanced data.

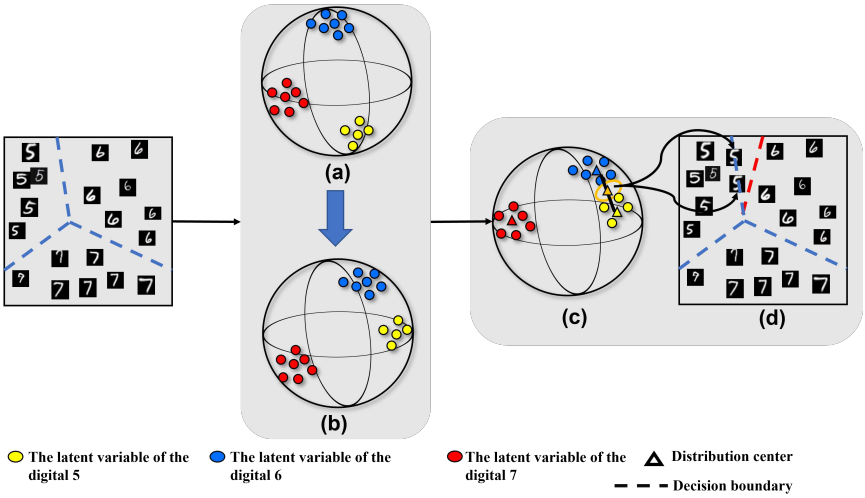


Fig. 5 An example of the proposed method.

In the early stages of training, each class forms a different vMF distribution (as shown in Fig. 5(a)) by using the method in subsection 3.1.1. Then, under the guidance of DM loss in subsection 3.1.2, the similar classes gradually get closer (as shown in Fig. 5(b), the yellow circle gradually moves to the blue circle). The generator can learn shared latent features from latent variables between similar classes. After the training converges, the sampling method in subsection 3.2 is adopted, which forms a sampling region between the red and the blue circle (as shown in Fig 5(c), the orange circle), and the parameter α can be used to adjust the position of the sampling region. Finally, latent variables are sampled from this region to generate boundary samples, thereby repairing the skewed decision boundary (as shown in Fig.5(d), the blue dotted line is moved to the red dotted line).

4 Experimental study

4.1 Datasets

The proposed method is evaluated on four publicly benchmark datasets, including MNIST [44], FASHION-MNIST [45], CIFAR-10 [46], CINICI-10 [47].

MNIST is a dataset used for digital image classification, consisting of grayscale handwritten digital images of 28x28 pixel size. MNIST has 10 categories corresponding to the numbers 0-9, including 50,000 training and 10,000 test data.

FASHION-MNIST is an alternative dataset to MNIST, containing 60,000 training samples and 10,000 test samples. Each sample is a 28x28 grayscale image associated with a label from 10 classes. These labels include T-shirts, pants, jumpers, dresses, coats, sandals, shirts, sneakers, bags, ankle boots.

CIFAR-10 is made up of 60,000 32x32 pixel colour images. This dataset has 50,000 training samples and 10,000 testing samples, divided into 10 classes: airplanes, cars, birds, cats, deer, dogs, frogs, horses, boats, and trucks.

CINIC-10 is constructed from CIFAR-10 and ImageNet. This dataset also consists of 32x32 colour images in 10 classes. Both training samples and test samples are 90,000. It fills the gap from CIFAR-10 to ImageNet.

The step imbalance and long-tailed imbalance datasets are created based on the experimental settings of Buda et al. [15] and Mullick et al. [30], respectively. These two versions of imbalanced data are achieved by subsampling the original dataset. An example of step imbalance and long-tailed imbalance is shown in Fig.6.

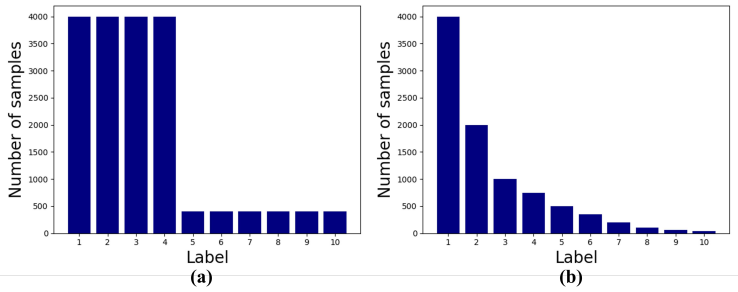


Fig. 6 Example distributions for imbalanced datasets (a) step imbalance, (b) long-tailed imbalance

There are two influencing factors to consider before creating an imbalanced dataset. The first factor is the Imbalance Ratio (IR) between the majority and the minority classes, which determines the learning difficulty of imbalanced problems. Therefore, it is necessary to set several sets of IR for each dataset to verify our method's performance comprehensively. The expression of IR is shown in Eq.(17). The second factor is the choice of the majority class. The classifier can easily learn discriminative features if there is a clear boundary between the majority and minority classes. However, if the boundary is not clear, it is another case. As a result, a different class should be chosen as the

majority class in the experiment.

$$IR = \frac{\text{The number of the majority class samples}}{\text{The number of the minority class samples}} \tag{17}$$

Based on the first factor, IR is set to 10, 30, and 50 for all datasets. Based on the second factor, different classes should be choosen as majority class and head class until every class is chosen. All possible combinations are tested using the original test set and then reporting the average test metric. Table 1 and Table 2 summarize the different versions of the imbalanced dataset.

Table 1 Overview of the step imbalance dataset used in our experiments

Dataset	IR	Majority	Minority	Dimension
MNIST	10,30,50	4000	400,133,80	28 x 28
FASHION-MNIST	10,30,50	4000	400,133,80	28 x 28
CIFAR-10	10,30,50	4500	450,150,90	32 x 32
CINIC-10	10,30,50	4500	450,150,90	32 x 32

Table 2 Overview of the long-tailed imbalance dataset used in our experiments

Dataset	IR	Training samples	Dimension
MNIST	100	4000,2000,1000,750,500,350,200,100,60,40	28 x 28
FASHION-MNIST	100	4000,2000,1000,750,500,350,200,100,60,40	28 x 28
CIFAR-10	100	4500,2000,1000,800,600,500,400,250,150,45	32 x 32
CINIC-10	100	4500,2000,1000,800,600,500,400,250,150,45	32 x 32

4.2 Compared methods

The performance of the proposed method is evaluated by comparison with the following five state-of-the-art methods. (1)B-SMOTE [18] synthesizes new minority class samples near the decision boundary. (2)ADASYN [21], an adaptive synthesis method of minority class samples. It assigns different weights to samples with different learning difficulties, and this method aims to generate more samples that are difficult to learn. (3)DVVAN [32], a network based on VAE-GAN. It generates boundary samples by modelling latent variables of the opposite mean for similar classes. (4)GAMO [30] designed a convex generator structure and then generated boundary samples through an adversarial game between the classifier and the convex generator. (5)ACGAN [39], a network based on GAN. It improves the quality of generated samples by embedding classifiers at the last layer of the discriminator.

4.3 Evaluation metrics

The most commonly used metric for evaluating classifier performance is overall accuracy. However, if the data is imbalanced, certain representative classes will lead to a highly misleading assessment. For example, there are 99 majority class samples and 1 minority class sample in a binary classification problem. When the classifier predicts that all samples are majority class, the classifier can achieve an accuracy of 99%, but the error rate of minority class is 100%. As a result, Precision, Recall, G-mean, and F1-score are used as evaluation metrics to evaluate the performance of the classifier. They are listed in Table 3, where TP stands for true positive, TN stands for true negative, FP stands for false positive, and FN stands for false negative, respectively. Precision represents

Table 3 The evaluation metrics used in experiment

metrics	Calculation
Precision	$\frac{TP}{TP+FP}$
Recall	$\frac{TP}{TP+FN}$
F1-score	$\frac{2TP}{2TP+FP+FN}$
G-mean	$\sqrt{\text{Recall} * \frac{TN}{TN+FP}}$

how many of the predicted true positive samples are actually positive samples. Recall reflects the proportion of samples predicted to be true positives and samples that are actually positive. F1-score is the weighted average of Precision and Recall. G-mean (geometric mean score) tries to maximize the accuracy on each of the classes while keeping these accuracies balanced.

4.4 Experimental setup

For a fair comparison, all models use the same number of network layers on the same dataset. Appendix A provides the detailed network architectures. B-SMOTE and ADASYN are implemented using the imbalanced learning library [48]. LeNet-5 [44] is used as the classification model on all datasets. The optimizer uses the stochastic gradient descent (SGD) with momentum value of $\mu = 0.9$, and the learning rate is 0.01.

When training the models, different datasets set different parameters. For MNIST and FASHION-MNIST datasets, DVVAN, GAMO, ACGAN and SCN are trained for 100 epochs, each batch size is 128. The optimizer uses Adam with the parameters $\beta_1 = 0$, $\beta_1 = 0.999$ and the learning rate is 0.0001. For CIFAR-10 and CINIC-10 datasets, the epoch and the batch are increased to 300 and 256, respectively. The learning rate is 0.0002.

4.5 Comparative Experiment

First train a classifier on the original imbalanced dataset, forming a Baseline for comparing model performance. Then using the above method and SCN to augment the imbalanced dataset, training the classifier again with the augmented dataset. lastly, comparing the classifier performance on the four datasets.

4.5.1 Results on step imbalance datasets

Table 4-5 reports the classification performance of representative methods on the step imbalance MNIST and FASHION-MNIST datasets with IR of 10, 30, 50, respectively. The comparison results show that when the IR is 10, the performance of all methods has only a small difference and outperform the Baseline. As the IR increases to 30, the different methods start to show a gap. Since ACGAN lacks mechanisms to handle imbalanced datasets, it slips below the Baseline, while other methods still outperform the Baseline. When the IR is further increased to 50, ACGAN and Baseline already lag behind other methods significantly. While our proposed method is able to generate samples near the decision boundary by using a boundary latent variable sampling strategy, and DM loss can ensure that the SCN generates samples of the correct class under extreme imbalance, which makes SCN outperform all other methods.

Table 4 Results of classification performance on the step imbalance MNIST

	IR=10				IR=30				IR=50			
Methods	Prec.	Rec.	F1.	GM.	Prec.	Rec.	F1.	GM.	Prec.	Rec.	F1.	GM.
Baseline	0.9640	0.9629	0.9630	0.9732	0.9388	0.9329	0.9333	0.9523	0.9220	0.9110	0.9214	0.9304
B-SMOTE	0.9675	0.9663	0.9636	0.9712	0.9451	0.9409	0.9413	0.9568	0.9234	0.9220	**0.9355**	0.9355
ADASYN	0.9668	0.9657	0.9659	0.9704	0.9436	0.9417	0.9404	0.9518	09272	0.9206	0.9222	0.9353
ACGAN	0.9661	0.9649	0.9650	0.9721	0.9221	0.9172	0.9161	0.9533	0.8879	0.8677	0.8632	0.9247
GAMO	**0.9732**	0.9682	0.9667	0.9783	0.9529	0.9473	0.9479	0.9691	0.9361	0.9343	0.9334	0.9460
DVVAN	0.9653	0.9637	0.9641	0.9779	0.9452	0.9403	0.9410	0.9565	0.9243	0.9081	0.9103	0.9382
SCN	0.9711	**0.9700**	**0.9703**	**0.9833**	**0.9536**	**0.9497**	**0.9504**	**0.9718**	**0.9387**	**0.9396**	0.9222	**0.9553**

Table 5 Results of classification performance on the step imbalance FASHION-MNIST

	IR=10				IR=30				IR=50			
Methods	Prec.	Rec.	F1.	GM.	Prec.	Rec.	F1.	GM.	Prec.	Rec.	F1.	GM.
Baseline	0.8495	0.8047	0.7810	0.8872	0.7701	0.7355	0.6789	0.8449	0.7099	0.7238	0.6641	0.7876
B-SMOTE	0.8557	0.8441	0.8404	0.9016	0.8189	0.7907	0.7643	0.8687	0.7742	0.7506	0.7351	0.8465
ADASYN	0.8555	0.8371	0.8318	0.9066	0.8267	0.8005	0.7878	0.8747	0.7783	0.7694	0.7376	0.8372
ACGAN	0.8496	0.8170	0.8088	0.8943	0.8011	0.7647	0.7303	0.8444	0.7474	0.7318	0.6963	0.8210
GAMO	0.8668	0.8534	0.8517	0.9144	0.8413	0.8254	0.8115	0.8996	0.8107	0.7860	0.7711	0.8759
DVVAN	0.8454	0.8144	0.8296	0.8930	0.8356	0.8193	0.7960	0.8610	0.7869	0.7551	0.7373	0.8425
SCN	**0.8708**	**0.8577**	**0.8581**	**0.9156**	**0.8531**	**0.8345**	**0.8192**	**0.9033**	**0.8324**	**0.8027**	**0.7921**	**0.8902**

Table 6-7 show the experimental results of the comparative methods on CIFAR-10 and CINIC-10. Compared with the traditional methods B-SMOTE and ADASYN, our method consistently outperforms them, while their performance drops sharply and falls below the Baseline. Because they generate samples that lack diversity and have more artifacts, which can be seen in B. Among the deep learning-based methods, ACGAN may suffer from

the problem of mode collapse, so its performance is poor. Compared with DVVAN, which is also modelling latent distributions, our method can generate high-quality samples through end-to-end learning, so the performance of the proposed method is superior to it. Furthermore, compared to GAMO, our method still achieves the highest score and make a large improvement on G-mean. Furthermore, comparative experiments were conducted under a variety

Table 6 Results of classification performance on the step imbalance CIFAR-10

	IR=10				IR=30				IR=50			
Methods	Prec.	Rec.	F1.	GM.	Prec.	Rec.	F1.	GM.	Prec.	Rec.	F1.	GM.
Baseline	0.5464	0.4562	0.4346	0.6547	0.5094	0.3800	0.3154	0.5848	0.4869	0.3535	0.2721	0.5227
B-SMOTE	0.4911	0.4041	0.3752	0.6142	0.4485	0.3348	0.2674	0.5568	0.4765	0.3187	0.2362	0.5227
ADASYN	0.5156	0.4349	0.4154	0.6384	0.4737	0.3472	0.2894	0.5674	0.47619	0.3279	0.2473	0.5208
ACGAN	0.5369	0.4293	0.4112	0.6152	0.4256	0.3719	0.3467	0.5520	0.3858	0.3531	0.3217	0.4645
GAMO	0.5671	0.5102	0.4561	0.6745	0.5193	0.4210	0.4016	0.6103	0.4922	0.3807	0.3469	0.5580
DVVAN	0.5506	0.5122	0.4537	0.6658	0.4804	0.3787	0.3051	0.5687	0.4877	0.3622	0.3176	0.5358
SCN	**0.5759**	**0.5215**	**0.4691**	**0.6932**	**0.5316**	**0.4910**	**0.4108**	**0.6378**	**0.5134**	**0.4244**	**0.3687**	**0.6077**

Table 7 Results of classification performance on the step imbalance CINIC-10

	IR=10				IR=30				IR=50			
Methods	Prec.	Rec.	F1.	GM.	Prec.	Rec.	F1.	GM.	Prec.	Rec.	F1.	GM.
Baseline	0.4134	0.3520	0.3139	0.5715	0.3626	0.3088	0.2269	0.5340	0.3493	0.2843	0.1787	0.4415
B-SMOTE	0.3803	0.3257	0.2855	0.5489	0.3406	0.2870	0.2090	0.5141	0.3256	0.2758	0.1814	0.4236
ADASYN	0.3806	0.3247	0.2843	0.5481	0.3425	0.2878	0.2090	0.5148	0.3219	0.2783	0.1863	0.4359
ACGAN	0.3876	0.2946	0.2712	0.5211	0.3368	0.2736	0.2387	0.5015	0.2943	0.2712	0.2450	0.4492
GAMO	0.4551	0.3873	0.3555	0.6060	0.3919	0.3472	0.3385	0.5675	0.3607	0.3111	0.3036	0.5022
DVVAN	0.4174	0.3413	0.3140	0.5824	0.3813	0.3009	0.3226	0.5467	0.3621	0.3054	0.2986	0.4896
SCN	**0.4638**	**0.3945**	**0.3670**	**0.6078**	**0.4344**	**0.3681**	**0.3512**	**0.5851**	**0.3812**	**0.3323**	**0.3208**	**0.5612**

of IR, thereby studying the trend of performance change with IR variation. To avoid performance degradation due to complex data overriding the role of IR, FASHION-MNIST is chosen for this study.

As shown in Fig.7 the proposed method consistently outperforms other methods. B-SMOTE and ADASYN have strong competitiveness until IR reaches 40. ACGAN is most severely affected by IR, and its performance lags behind other methods when IR comes 30. Both our method and DVVAN model the latent variables of similar classes. Our method chooses different similar classes by using a pre-training classification model. However, DVVAN relies on subjective judgment to choose similar classes, so it may not be able to reshape the correct class boundary if it chooses inappropriate classes, which is the reason for its significant performance fluctuations. In addition, our method and GAMO can effectively alleviate the imbalanced problem.

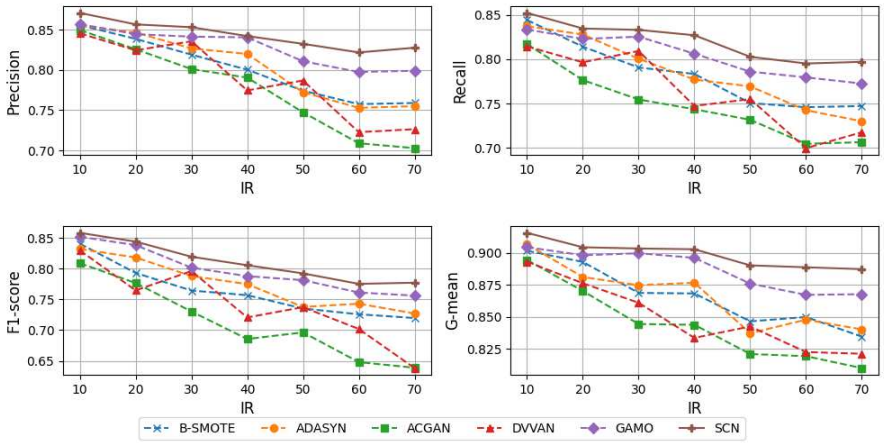


Fig. 7 The performance of representative methods varies with IR.

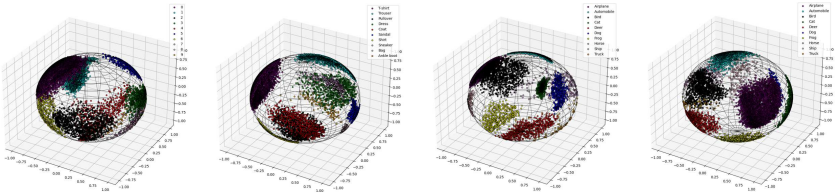
4.5.2 Results on long-tailed imbalance datasets

Table 8 shows the experimental results of representative methods on long-tailed imbalance datasets. Each class has a different IR from other classes in long-tail datasets, so the decision boundary is more complex. Therefore, it can be observed that the performance of ACGAN on low-dimensional datasets has fallen behind the Baseline. Compared with other methods, our method achieves better performance because SCN model similar classes based on misclassification probability, that the generated samples tend to near the decision boundary of two easily misclassified classes, so that the classifier is able to learn a more robust class Boundary. Experimental results demonstrate that our method is effective in long-tailed datasets with more complex classification boundaries.

Table 8 Results of classification performance on long-tailed imbalance datasets

	MNIST				Fashion-MNIST				CIFAR-10				CINIC-10			
Methods	Prec.	Rec.	F1.	GM.	Prec.	Rec.	F1.	GM.	Prec.	Rec.	F1.	GM.	Prec.	Rec.	F1.	GM.
Baseline	0.9297	0.9258	0.9245	0.9582	0.8078	0.7802	0.7518	0.8724	0.5175	0.4034	0.3657	0.6137	0.4215	0.2821	0.2005	0.5095
B-SMOTE	0.9361	0.9342	0.9329	0.9630	0.8241	0.8108	0.8060	0.8909	0.4735	0.3988	0.3689	0.6100	0.3119	0.2986	0.2566	0.5247
ADASYN	0.9365	0.9346	0.9334	0.9632	0.8268	0.7896	0.7808	0.8781	0.4628	0.3913	0.3627	0.6040	0.3158	0.2973	0.2555	0.5235
ACGAN	0.9095	0.8998	0.8975	0.9434	0.8079	0.7687	0.7425	0.8521	0.3381	0.3551	0.3345	0.5741	0.2638	0.2774	0.2595	0.5051
GAMO	0.9497	0.9490	0.94899	0.9671	0.8422	0.8334	0.8273	0.9044	0.5292	0.4788	0.4680	0.6716	0.4484	0.3548	0.3430	0.5739
DVVAN	0.9409	0.9395	0.9280	0.9603	0.8226	0.8083	0.7916	0.8934	0.5035	0.4098	0.3822	0.6281	0.4266	0.3370	0.3107	0.5241
SCN	**0.9541**	**0.9521**	**0.9512**	**0.9675**	**0.8533**	**0.8471**	**0.8351**	**0.9347**	**0.5573**	**0.5156**	**0.4907**	**0.7048**	**0.4707**	**0.4410**	**0.4256**	**0.6326**

Furthermore, Fig.8 shows the visualization of SCN modeling results on long-tailed datasets. In the latent space, each class has a clear boundary, and similar classes are close to each other. The encoder shows good modelling performance, except for some outlier latent variables.



(a) MNIST (b) FMNIST (c) CIFAR-10 (d) CINIC-10

Fig. 8 Modelling results of different datasets in latent space.

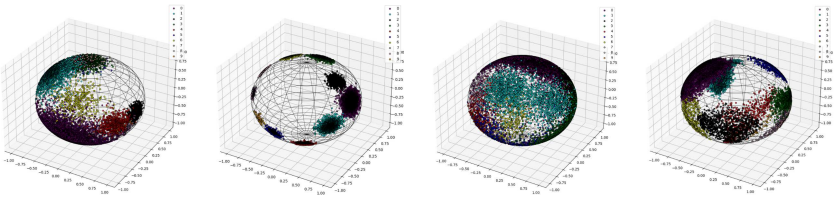
4.6 Ablation study

This section conducts ablation experiments on DM loss and boundary sampling strategy.

4.6.1 Ablation study for DM loss

In order to verify the role of DM loss, we created three ablation entities. 1)SCN-A removes DM loss from the encoder and classification loss from the generator. 2)SCN-B removes DM loss from the encoder. 3)SCN-C removes classification loss from the generator. The experimental results are shown in Table 9, SCN outperforms all other ablation variants.

The modelling results are visualised to illustrate the role of DM loss better. As shown in Fig.9, SCN-A only models the latent distribution in terms of reconstruction loss, thus forming many overlapping clusters under the influence of imbalanced data. Since the lack of DM loss in SCN-B, it can only model each class as a different latent distribution. Compared with SCN-B, SCN-C is the complete opposite, it models similar classes closer, but it causes the overlap between class distributions. In the modelling results of SCN, similar classes are closer to each other, and the class boundary is distinguishable. Therefore, the classification loss and DM loss are complementary relationships.



(a) SCN-A (b) SCN-B (c) SCN-C (d) SCN

Fig. 9 modelling results of three variants and SCN.

Table 9 Results of Ablation study for DM loss on MNIST dataset

	Long-tailed imbalance				Step imbalance (IR=30)				Step imbalance (IR=50)			
Methods	Prec.	Rec.	F1.	G-mean	Prec.	Rec.	F1.	G-mean	Prec.	Rec.	F1.	G-mean
Baseline	0.9266	0.9222	0.9206	0.9470	0.9370	0.9340	0.9295	0.9558	0.9206	0.9079	0.9095	0.9380
SCN-A	0.8684	0.8752	0.8848	0.9213	0.8784	0.8553	0.8629	0.8677	0.8307	0.8492	0.8398	0.8492
SCN-B	0.9252	0.9234	0.9341	0.9481	0.9143	0.92064	0.9303	0.9481	0.9223	0.9254	0.9292	0.9401
SCN-C	0.8712	0.8852	0.8812	0.9331	0.8712	0.8664	0.8692	0.8840	0.8432	0.8588	0.8419	0.8531
SCN	**0.9598**	**0.9541**	**0.9588**	**0.9616**	**0.9548**	**0.9525**	**0.9221**	**0.9707**	**0.9350**	**0.9268**	**0.9272**	**0.9588**

4.6.2 Ablation study for sampling strategy

Many existing deep generative model-based methods generate samples by sampling distribution centers. As a result, to demonstrate that our sampling strategy contributes more to training a classifier, an experiment was designed to compare different sampling centers. We create an ablation variant (SCN-Center) that takes the distribution center of each class as the sampling center.

The experimental results are listed in Table 10, SCN-Center outperforms Baseline in terms of generating samples of the correct class to augment the imbalanced dataset. However, the samples generated by SCN-Center are far away from the classification boundary, which has limited contribution to the classifier learning a robust classification boundary, so its performance lags behind that of SCN

Table 10 Results of Ablation study for sampling strategy on MNIST dataset

	Long-tailed imbalance				Step imbalance (IR=30)				Step imbalance (IR=50)			
Method	Prec.	Rec.	F1.	G-mean	Prec.	Rec.	F1.	G-mean	Prec.	Rec.	F1.	G-mean
Baseline	0.9256	0.9241	0.9233	0.9310	0.9310	0.9292	0.9203	0.9401	0.9196	0.9094	0.9103	0.9389
SCN-Center	0.9332	0.9342	0.9412	0.9441	0.9382	0.9323	0.9357	0.9462	0.9186	0.9183	0.9154	0.9377
SCN	**0.9541**	**0.9522**	**0.9574**	**0.9676**	**0.9589**	**0.9563**	**0.9556**	**0.9755**	**0.9447**	**0.9393**	**0.9399**	**0.9559**

5 Conclusion and future work

This paper proposes a similar classes latent distribution modelling-based oversampling method to alleviate the difficulty of learning from imbalanced image datasets. First, we model each class as a different vMF distribution to reduce the difficulty of learning from a unimodal distribution. Second, a distance minimization loss function is introduced in the encoder, which makes similar classes closer, so that the generator can learn shared latent features in the decision region of similar classes. In addition, similar classes are selected using a pre-trained classifier, which can effectively avoid the bias caused by human selection. The classification model is trained with imbalanced datasets, which can further reflect the skewed direction of the decision boundary. Therefore, our method can fix skewed class boundaries more targeted. Finally, we design a boundary sampling strategy, which can sample latent variables in the decision region to generate boundary samples. By adding these samples to the training set, the classifier can learn more difficult-to-classify (similar) features and

further improve the robustness of the classifier to imbalanced image datasets. In extensive experiments, it has been demonstrated that the proposed method can effectively handle the imbalance problem. On CIFAR-10 and CINIC-10 with step imbalance (IR=50), G-mean improves by about 5% and 6%, respectively. Furthermore, G-mean is enhanced by about 3% and 5% on CIFAR-10 and CINIC-10 with long-tailed imbalance, respectively.

In future work, we consider combining some manifold learning methods to obtain the manifold structure of the training data, thereby selecting similar classes more accurately. Besides, we plan to apply our method to real-world scenarios instead of benchmark datasets.

Acknowledgments. This work is supported by the National Natural Science Foundation of China (No. 61563012), the Guangxi Natural Science Foundation of China (No.2021GXNSFAA220074), and the Guangxi Key Laboratory of Embedded Tech-nology and Intelligent System Foundation (No.2019-1-4).

Declarations

Ethical Approval. not applicable

Competing interests. The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Authors' contributions. Wei Ye: Writing - original draft, Methodology, Validation. Minggang Dong: Conceptualization, Writing - review & editing, Supervision, Funding acquisition. Yan Wang: Validation, Coding. Guojun Gan: Validation. Deao Liu: Coding

Funding. This work is supported by the National Natural Science Foundation of China (No. 61563012), the Guangxi Natural Science Foundation of China (No.2021GXNSFAA220074), and the Guangxi Key Laboratory of Embedded Tech-nology and Intelligent System Foundation (No.2019-1-4).

Availability of data and materials. The datasets generated and analysed during the current study are available from the corresponding author on reasonable request.

Appendix A Network architectures

Table A1 The network structure of the classification model LeNet-5

Layer	Activation	Kernel size	Dimension
Input	-	-	32x32x1
Convolution	ReLu	5	28x28x1
Max pooling	-	2	14x14x1
Convolution	ReLu	5	10x10x1
Max pooling	-	2	5x5x1
Fully connected	ReLu	-	120
Fully connected	ReLu	-	84
Fully connected		-	10
SoftMax	-	-	10

Table A2 The network architecture of Generator for MNIST, FASHION-MNIST

Layer	Activation	Kernel size	Channel
Input	-	-	20
Fully connected	ReLu	-	1024
Fully connected	ReLu	-	128
Convolution	ReLu	4	64
ConvTranspose	ReLu	4	3

Table A3 The network architecture of Discriminator for MNIST, FASHION-MNIST

Layer	Activation	Kernel size	Channel
Input	-	-	20
Fully connected	ReLu	-	1024
Fully connected	ReLu	-	128
Convolution	ReLu	4	64
ConvTranspose	ReLu	4	1

Table A4 The network architecture of Encoder for MNIST, FASHION-MNIST

Layer	Activation	Kernel size	Channel
Input	-	-	1
Fully connected	ReLu	-	400
Fully connected	-	-	3

Table A5 The network architecture of Generator for CIFAR-10, CINIC-10

Layer	Activation	Kernel size	Channel
Input	-	-	40
ConvTranspose	ReLu	4	512
ConvTranspose	ReLu	4	256
ConvTranspose	ReLu	4	128
ConvTranspose	ReLu	4	64
ConvTranspose	ReLu	1	3

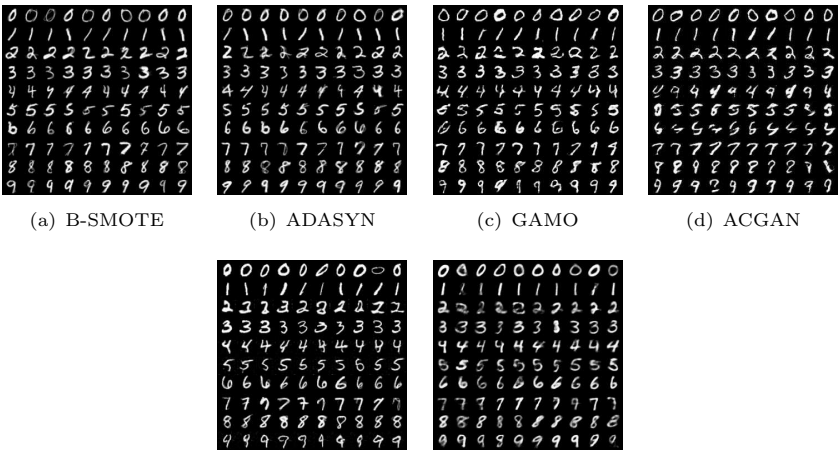
Table A6 The network architecture of Discriminator for CIFAR-10, CINIC-10

Layer	Activation	Kernel size	Channel
Input	-	-	3
Convolution	lReLu	4	64
ConvTranspose	lReLu	4	128
ConvTranspose	lReLu	4	256
ConvTranspose	lReLu	4	512
ConvTranspose	Sigmoid	1	1

Table A7 The network architecture of Encoder for CIFAR-10, CINIC-10

Layer	Activation	Kernel size	Channel
Input image	-	-	3
Convolution	ReLu	4	64
Convolution	ReLu	4	128
Convolution	ReLu	4	256
Fully connected	ReLu	-	2048
Fully connected	-	-	3

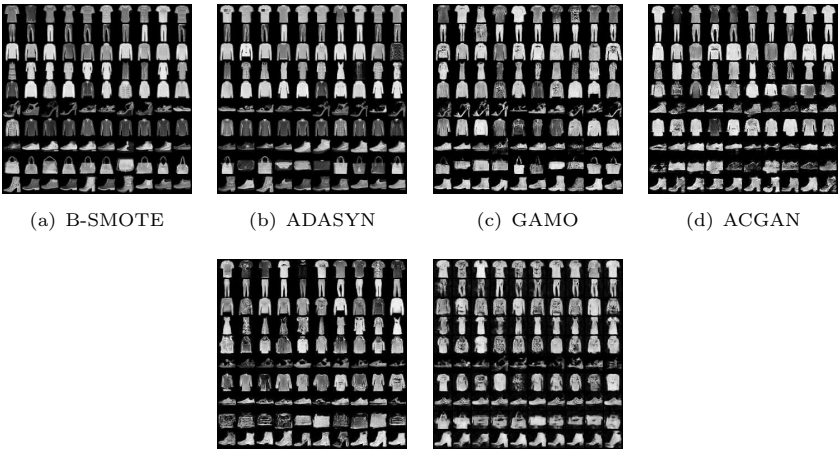
Appendix B Examples of generated images



(a) B-SMOTE (b) ADASYN (c) GAMO (d) ACGAN

(e) SCN (f) DVVAN

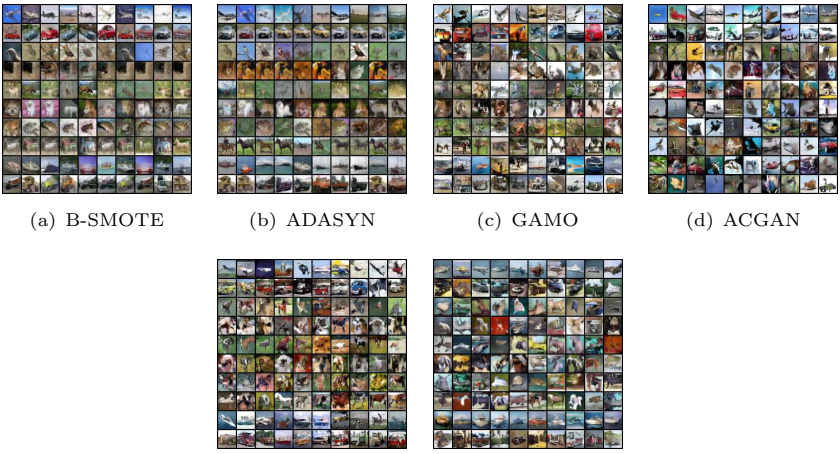
Fig. B1 The generated images of different oversampling approaches on MNIST dataset.



(a) B-SMOTE (b) ADASYN (c) GAMO (d) ACGAN

(e) SCN (f) DVVAN

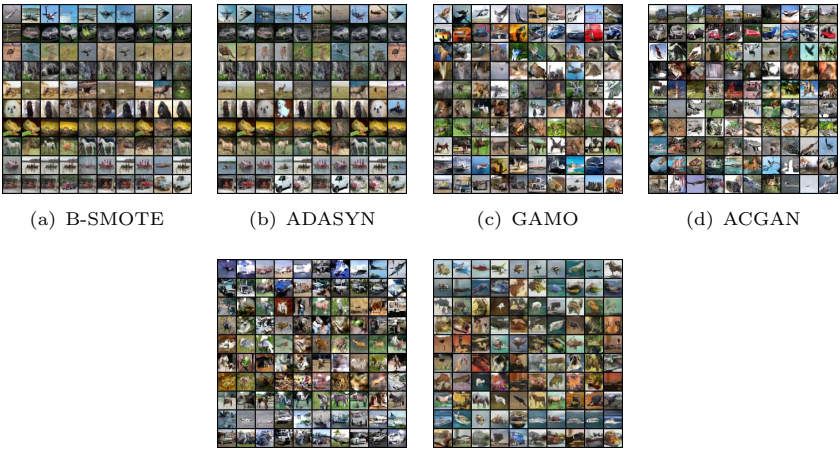
Fig. B2 The generated images of different oversampling approaches on FASHION-MNIST dataset.



(a) B-SMOTE (b) ADASYN (c) GAMO (d) ACGAN

(e) SCN (f) DVVAN

Fig. B3 The generated images of different oversampling approaches on CIFAR-10 dataset.



(a) B-SMOTE (b) ADASYN (c) GAMO (d) ACGAN

(e) SCN (f) DVVAN

Fig. B4 The generated images of different oversampling approaches on CINIC-10 dataset.

References

[1] Zhou, B., Cui, Q., Wei, X.-S., Chen, Z.-M.: Bbn: Bilateral-branch network with cumulative learning for long-tailed visual recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9719–9728 (2020)

[2] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., *et al.*: Imagenet large scale visual recognition challenge. International journal of computer vision

115(3), 211–252 (2015)

[3] Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: European Conference on Computer Vision, pp. 740–755 (2014). Springer

[4] Wang, J., Lukasiewicz, T., Hu, X., Cai, J., Xu, Z.: Rsg: A simple but effective module for learning imbalanced datasets. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 3784–3793 (2021)

[5] Khan, S.H., Hayat, M., Bennamoun, M., Sohel, F.A., Togneri, R.: Cost-sensitive learning of deep feature representations from imbalanced data. IEEE transactions on neural networks and learning systems **29**(8), 3573–3587 (2017)

[6] Krawczyk, B.: Learning from imbalanced data: open challenges and future directions. Progress in Artificial Intelligence **5**(4), 221–232 (2016)

[7] He, H., Garcia, E.A.: Learning from imbalanced data. IEEE Transactions on knowledge and data engineering **21**(9), 1263–1284 (2009)

[8] Catania, C.A., Bromberg, F., Garino, C.G.: An autonomous labeling approach to support vector machines algorithms for network traffic anomaly detection. Expert Systems with Applications **39**(2), 1822–1829 (2012)

[9] Reza, M.S., Ma, J.: Imbalanced histopathological breast cancer image classification with convolutional neural network. In: 2018 14th IEEE International Conference on Signal Processing (ICSP), pp. 619–624 (2018). IEEE

[10] Jain, A., Ratnoo, S., Kumar, D.: Addressing class imbalance problem in medical diagnosis: A genetic algorithm approach. In: 2017 International Conference on Information, Communication, Instrumentation and Control (ICICIC), pp. 1–8 (2017). IEEE

[11] Li, X., Li, K.: High-dimensional imbalanced biomedical data classification based on p-adaboost-pauc algorithm. The Journal of Supercomputing, 1–24 (2022)

[12] Lin, T.-Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2980–2988 (2017)

[13] Fajardo, V.A., Findlay, D., Jaiswal, C., Yin, X., Houmanfar, R., Xie, H., Liang, J., She, X., Emerson, D.: On oversampling imbalanced data with

deep conditional generative models. Expert Systems with Applications **169**, 114463 (2021)

[14] Wang, X., Xu, J., Zeng, T., Jing, L.: Local distribution-based adaptive minority oversampling for imbalanced data classification. Neurocomputing **422**, 200–213 (2021)

[15] Buda, M., Maki, A., Mazurowski, M.A.: A systematic study of the class imbalance problem in convolutional neural networks. Neural networks **106**, 249–259 (2018)

[16] Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: Smote: synthetic minority over-sampling technique. Journal of artificial intelligence research **16**, 321–357 (2002)

[17] Suh, S., Lee, H., Lukowicz, P., Lee, Y.O.: Cegan: Classification enhancement generative adversarial networks for unraveling data imbalance problems. Neural Networks **133**, 69–86 (2021)

[18] Han, H., Wang, W.-Y., Mao, B.-H.: Borderline-smote: a new over-sampling method in imbalanced data sets learning. In: International Conference on Intelligent Computing, pp. 878–887 (2005). Springer

[19] Bunkhumpornpat, C., Sinapiromsaran, K., Lursinsap, C.: Safe-level-smote: Safe-level-synthetic minority over-sampling technique for handling the class imbalanced problem. In: Pacific-Asia Conference on Knowledge Discovery and Data Mining, pp. 475–482 (2009). Springer

[20] Barua, S., Islam, M.M., Yao, X., Murase, K.: Mwmote–majority weighted minority oversampling technique for imbalanced data set learning. IEEE Transactions on knowledge and data engineering **26**(2), 405–425 (2012)

[21] He, H., Bai, Y., Garcia, E.A., Li, S.: Adasyn: Adaptive synthetic sampling approach for imbalanced learning. In: 2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence), pp. 1322–1328 (2008). IEEE

[22] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. Advances in neural information processing systems **27** (2014)

[23] Kingma, D.P., Welling, M.: Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114 (2013)

[24] Liu, J., Gu, C., Wang, J., Youn, G., Kim, J.-U.: Multi-scale multi-class conditional generative adversarial network for handwritten character generation. The Journal of Supercomputing **75**(4), 1922–1940 (2019)

[25] Douzas, G., Bacao, F.: Effective data generation for imbalanced learning using conditional generative adversarial networks. Expert Systems with applications **91**, 464–471 (2018)

[26] Antoniou, A., Storkey, A., Edwards, H.: Data augmentation generative adversarial networks. arXiv preprint arXiv:1711.04340 (2017)

[27] Islam, Z., Abdel-Aty, M., Cai, Q., Yuan, J.: Crash data augmentation using variational autoencoder. Accident Analysis & Prevention **151**, 105950 (2021)

[28] Ali-Gombe, A., Elyan, E.: Mfc-gan: class-imbalanced dataset classification using multiple fake class generative adversarial network. Neurocomputing **361**, 212–221 (2019)

[29] Son, M., Jung, S., Jung, S., Hwang, E.: Bcgan: A cgan-based oversampling model using the boundary class for data balancing. The Journal of Supercomputing **77**(9), 10463–10487 (2021)

[30] Mullick, S.S., Datta, S., Das, S.: Generative adversarial minority oversampling. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 1695–1704 (2019)

[31] Choi, H.-S., Jung, D., Kim, S., Yoon, S.: Imbalanced data classification via cooperative interaction between classifier and generator. IEEE Transactions on Neural Networks and Learning Systems (2021)

[32] Guo, T., Zhu, X., Wang, Y., Chen, F.: Discriminative sample generation for deep imbalanced learning. In: Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19, pp. 2406–2412. International Joint Conferences on Artificial Intelligence Organization, ??? (2019)

[33] Park, S., Hong, Y., Heo, B., Yun, S., Choi, J.Y.: The majority can help the minority: Context-rich minority oversampling for long-tailed classification. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 6887–6896 (2022)

[34] Larsen, A.B.L., Sønderby, S.K., Larochelle, H., Winther, O.: Autoencoding beyond pixels using a learned similarity metric. In: International Conference on Machine Learning, pp. 1558–1566 (2016). PMLR

[35] Gurumurthy, S., Kiran Sarvadevabhatla, R., Venkatesh Babu, R.: Deligan: Generative adversarial networks for diverse and limited data. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 166–174 (2017)

[36] Fernando, K.R.M., Tsokos, C.P.: Dynamically weighted balanced loss: class imbalanced learning and confidence calibration of deep neural networks. IEEE Transactions on Neural Networks and Learning Systems (2021)

[37] Dong, Q., Gong, S., Zhu, X.: Imbalanced deep learning by minority class incremental rectification. IEEE transactions on pattern analysis and machine intelligence **41**(6), 1367–1381 (2018)

[38] Johnson, J.M., Khoshgoftaar, T.M.: Survey on deep learning with class imbalance. Journal of Big Data **6**(1), 1–54 (2019)

[39] Odena, A., Olah, C., Shlens, J.: Conditional image synthesis with auxiliary classifier gans. In: International Conference on Machine Learning, pp. 2642–2651 (2017). PMLR

[40] Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., Courville, A.C.: Improved training of wasserstein gans. Advances in neural information processing systems **30** (2017)

[41] Mariani, G., Scheidegger, F., Istrate, R., Bekas, C., Malossi, C.: Bagan: Data augmentation with balancing gan. arXiv preprint arXiv:1803.09655 (2018)

[42] Tanabe, A., Fukumizu, K., Oba, S., Takenouchi, T., Ishii, S.: Parameter estimation for von mises–fisher distributions. Computational Statistics **22**(1), 145–157 (2007)

[43] Liu, W., Wen, Y., Yu, Z., Li, M., Raj, B., Song, L.: Sphereface: Deep hypersphere embedding for face recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 212–220 (2017)

[44] LeCun, Y., Boser, B., Denker, J., Henderson, D., Howard, R., Hubbard, W., Jackel, L.: Handwritten digit recognition with a back-propagation network. Advances in neural information processing systems **2** (1989)

[45] Xiao, H., Rasul, K., Vollgraf, R.: Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. arXiv preprint arXiv:1708.07747 (2017)

[46] Krizhevsky, A., Nair, V., Hinton, G.: Cifar-10 (canadian institute for advanced research). URL http://www. cs. toronto. edu/kriz/cifar. html **5**(4), 1 (2010)

[47] Darlow, L.N., Crowley, E.J., Antoniou, A., Storkey, A.J.: Cinic-10 is not imagenet or cifar-10. arXiv preprint arXiv:1810.03505 (2018)

[48] Lemaître, G., Nogueira, F., Aridas, C.K.: Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning. The Journal of Machine Learning Research **18**(1), 559–563 (2017)