A Class Imbalance Loss for Imbalanced Object Recognition

Linbin Zhang, Caiguang Zhang, Sinong Quan, Huaxin Xiao, Gangyao Kuang, *Senior Member, IEEE*, and Li Liu, *Senior Member, IEEE*

***Abstract*—The class imbalance problem exists widely in vision data. In these imbalanced datasets, the majority classes dominate the loss and influence the gradient. Hence, these datasets have a significantly negative impact on the performance of many state-of-the-art methods. In this article, we propose a class imbalance loss (CI loss) to handle this problem. To distinguish imbalanced datasets in accordance with the extent of imbalance, we also define an imbalance degree that works as a decision index factor in the CI loss. Because the minority classes with fewer samples probably lose chances in descending the gradient in the training process, CI loss is introduced to make these minority classes descend further than the majority classes. In view of the imbalanced distribution of data in few-shot learning, a method for generating an imbalanced few-shot learning dataset is presented in this article. We conducted a large number of experiments in the MiniImageNet dataset, which showed the effectiveness of an algorithm for model-agnostic meta-learning for rapid adaptation with CI loss. In the problem of detecting 15 ship categories, our loss function is transplanted to a rotational region convolutional neural network detection method and a cascade network architecture and achieves higher mean average precision than focal loss and cross-entropy loss. In addition, the Mixed National Institute of Standards and Technology dataset and the Moving and Stationary Target Acquisition and Recognition dataset are sampled to imbalance datasets to verify the effectiveness of CI loss.**

***Index Terms*—Convolutional neural networks (CNNs), few-shot learning, image classification, imbalanced learning, loss functions, object detection.**

I. Introduction

IN MANY domains, data, including visual data, naturally exhibit imbalance in their category distribution. These data are referred to as "class imbalanced data," where most of the data belong to a few majority categories, while many minority categories only contain several or a few samples [1]. A practical recognition system has to operate in an open world, continuously and simultaneously dealing with a very large set of domains, including recognizing thousands of different object categories whose frequency distribution is long tailed (e.g., the well-known ImageNet dataset [2]). A practical recognition system must handle class imbalanced data. However, the problem of learning from the class imbalanced dataset, i.e., the imbalanced learning problem, has been a challenging and longstanding problem [3]–[6]. Despite significant progress brought by deep learning [7]–[9], most of the existing deep learning methods consider class balanced datasets (such as the ImageNet1000 for image classification competition [10]) or moderately imbalanced datasets. Class imbalance, especially severe class imbalance, has a significantly negative impact on the performance of many state-of-the-art object detection and classification methods [11]–[13]. Recently, there have been some emerging attempts to address the challenge of learning from significantly skewed datasets [11]–[13]. Nevertheless, deep-learning-based class imbalance learning for visual recognition tasks is largely underexplored. In this article, our main focus is to investigate deep representation learning on class imbalanced data for object classification and detection problems.

Class imbalanced learning approaches intend to reduce the model learning bias toward majority categories by raising the importance of minority categories [12], [14], [15]. Existing methods for class imbalance handling in object recognition can be grouped into the following categories [1]: data-level, algorithm-level, and hybrid. However, most of them are ordinary imbalanced methods, and they are unable to deal with the highly imbalanced dataset [16]. In a general way, the imbalance ratio (IR) of the number of samples in maximum class and minimum class in a dataset is over 10:1 and can be regarded as the highly imbalanced learning, and the way to calculate the IR is shown in (1).

In class highly imbalanced learning, majority classes have the dominant effect during the learning process. In other words, the classification costs of the majority classes and minority classes are not equal, which will lead to the classifier preferring the majority class. During the deep neural network training process, when summed over a large number of training samples from the majority classes, these small loss values can overwhelm the rare class [13], [15]. That is to say, the majority classes comprise

Manuscript received September 24, 2019; revised February 3, 2020 and March 22, 2020; accepted May 15, 2020. Date of publication May 28, 2020; date of current version June 11, 2020. This work was supported in part by the National Natural Science Foundation of China under Grant 61872379, Grant 61701508, Grant 61372163, Grant 61906206, and Grant 71701205, and in part by the Hunan Provincial Natural Science Foundation of China under Grant 2018JJ3613. *(Corresponding author: Li Liu.)*

Linbin Zhang, Caiguang Zhang, Sinong Quan, and Gangyao Kuang are with the State Key Laboratory of Complex Electromagnetic Environment Effects on Electronics and Information System, National University of Defense Technology, Changsha 410073, China (e-mail: zlbnudt@163.com; zhangiguang@163.com; qsnong@hotmail.com; kuangyeats@hotmail.com).

Huaxin Xiao is with the College of System Engineering, National University of Defense Technology, Changsha 410073, China (e-mail: huaxinxiao@hotmail.com).

Li Liu is with the College of System Engineering, National University of Defense Technology, Changsha 410073, China, and also with the Center for Machine Vision and Signal Analysis, University of Oulu 90570, Oulu, Finland (e-mail: li.liu@oulu.fi).

Digital Object Identifier 10.1109/JSTARS.2020.2995703

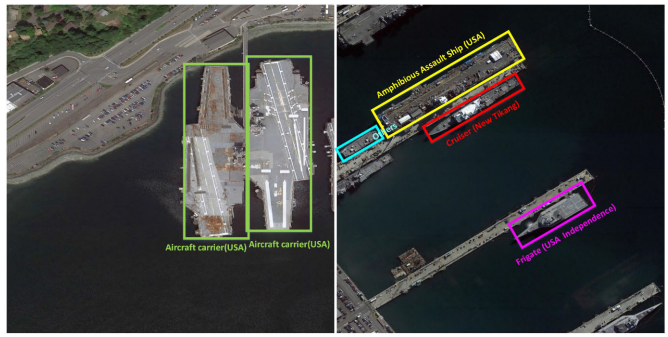
the majority of the loss and dominate the gradient. In order to mitigate the dominant effect of majority classes, we introduce a novel class imbalance loss (CI loss) that can be deployed readily in deep neural network architectures. We propose a novel loss that can downweight the gradient contribution of the samples in the majority classes and thus focus training on samples of minority classes.

To address this, in this article, a new loss function is proposed to alleviate the imbalance degree (ID) in the process of gradient descent, which can be used among arbitrary imbalanced datasets in applications such as classification and object detection. The loss function is modified on the basis of cross-entropy loss (CE loss). The inspiration for the new loss is from the concept of gradient descent. When training samples are equal among all classes, the number of samples from each class is mainly approximate to each other in a batch. The model and parameters will be trained to adapt to every class, until, eventually, the recognition rate achieves a high rate of accuracy. If the dataset is imbalanced, the situation will change, and the gradient will descend in the direction of recognizing the majority classes and lose the chance to identify the minority classes due to the imbalance. To make up for the absence of the minority, a factor λ is introduced in the CE loss, which will adjust the scale of gradient descent. The minority classes are able to decline further owing to the new loss. In addition, a dynamically scaled factor related to sin and cos is utilized in the loss, which demonstrates better performance in classification tasks. To control this factor λ, we define ID as a way of estimating the extent of imbalance of a specific dataset, which takes all classes into consideration. Using ID, we can constrain the factor λ of the imbalanced dataset.

To verify the effectiveness of our proposed CI loss, we define three tasks: imbalanced image classification, imbalanced few-shot image classification, and imbalanced optical ship detection and recognition. In imbalanced image classification, we sample some categories of datasets and conduct the contrasting experiments in both the optical image dataset and the synthetic aperture radar (SAR) image dataset. Furthermore, we analyze the change in the recognition rate with related ID, when the class number and the total amount of data are fixed in experiments involving MiniImageNet [17]. In addition, our proposed loss function is also used in the detection and recognition of 15 categories of optical ships. Experimental results on MiniImageNet, Mixed National Institute of Standards and Technology (MNIST) dataset, Moving and Stationary Target Acquisition and Recognition (MSTAR) dataset, and the optical ship location and identification dataset have shown that the proposed loss function performs better than other state-of-the-art methods.

Our contributions include the following.

1) We propose the CI loss for arbitrary imbalance datasets. This loss can be used in classification and object detection tasks, without adding more compute resources to our experiments. To limit the size of the factor in CI loss, we propose a scientific index to define the ID of arbitrary imbalance datasets, which takes the distribution of data in all classes into consideration, not just the maximum and minimum categories.



(a) (b)

Fig. 1. Ships with skew bounding box annotations. In (a), there are two aircraft carriers, which can be labeled in both skew and vertical rectangle boxes. In (b), ships should be labeled in skew boxes; otherwise, there will be areas containing useless or confusing information.

2) We embed our loss function into the total loss function in rotational region convolutional neural network (R2CNN) detection method [18] and the cascade network architecture [19], which means our loss function not only increases the recognition rate in classification tasks, but also performs well in object detection tasks based on locating and identifying ships in optical remote sensing images.
3) Experiments demonstrate that our proposed method can be easily adapted to classification and object detection tasks and that it outperforms other state-of-the-art approaches.

The rest of this article is composed of four sections. In Section II, we introduce the related work about object recognition, class imbalanced learning, and few-shot learning. In Section III-A, we describe the problem setup of imbalance datasets and ID, respectively. CI loss is presented in Section III-B, and the relationship between loss and gradient descent is theoretically proved in Section III-C. Subsequently, different applications of imbalance loss in the classification and object detection are introduced in Section IV. Experimental results and implementation details are summarized and presented in Section V. Finally, Section VI concludes this article.

II. Related Work

A. Object Classification and Detection

Object recognition is one of the most fundamental and challenging problems in computer vision. Object detection problems involve locating object instances from a large number of predefined categories in natural images. Deep learning techniques have emerged as strategies for learning feature representations directly from data and have led to breakthroughs in the field of generic object detection [8]. As deep learning evolves, methods based on convolutional neural networks (CNNs) have been presented, such as SSD [20], YOLO [21], R-CNN [22], Fast R-CNN [23], and Faster R-CNN [24]. Vertical rectangle bounding boxes, as shown in Fig. 1(a), have made tremendous progress in natural image datasets. However, when it comes to the problem of ship detection, rectangle bounding boxes will cover part of the sea or shoreside in targets. If a series of ships are compact

and parallel to each other in an inclination angle, the vertical rectangle bounding boxes will often include part of the adjacent ships, because ships are not always vertical or horizontal to the entire remote sensing image. As shown in Fig. 1(b), the amphibious assault ship (U.S., yellow rectangle) is slanted at a 30° angle compared to the horizontal lines, and if it is labeled with a vertical rectangle, then sea, shore, and even the entire cruiser (New Tikang, red rectangle) will be included in the bounding box. For this situation, deep learning methods for ship detection in arbitrary orientation have been proposed [25]–[27]. However, our dataset contains 15 ship categories of different sizes and functions, which is more intractable than common detection tasks. We chose R2CNN [18] and Cascade [19] as detection methods, which are initially used in orientation robust scene test detection, as the base detectors but with different parts in loss of classification.

Many researchers have worked hard on designing algorithms to solve classification problems. ImageNet [7] surpasses nearly all the traditional machine learning methods, and deep learning methods are used in fields related to computer vision, such as face recognition [28], handwriting recognition [29], and SAR image recognition [30]. The image classification datasets used in this article are MNIST and MSTAR. Because many methods have achieved over 99% recognition rate on the MNIST dataset [31]–[33], it is useful to contrast our loss function with other methods. In fact, SAR target recognition has been increasingly used in both civilian and military fields. In recent years, a number of methods have been proposed to face this challenge [30], [34]–[38]. Among these, a deep CNN has achieved 99.13% [30] in the MSTAR public dataset. However, researchers seldom pay attention to the imbalanced dataset of MSTAR. When it comes to data about noncooperative targets, the imbalanced SAR target dataset is frequently used. Some categories are selected to be minority classes, and CI loss is preferable than other losses, according to our experiments.

B. Imbalanced Learning

The aim of imbalanced learning is to reduce the bias of the model by increasing the significance of minority classes. To cope with the side effects of imbalance in datasets, many researchers investigate improvements and algorithms through trial and error. These methods include in data-level, algorithm-level, and hybrid methods. Some data-level methods are traditional methods in upsampling and downsampling, as described in SMOTE [39]–[41], or use generative neural networks [42] and other image processing improvements [43]–[45] as augmentation to minority classes. Algorithm-level methods mainly increase the importance of the minority classes by taking a class penalty into consideration instead of altering the distribution of training data. Among these methods, cost-sensitive learning [46] involves a cost matrix to increase the importance in minority class or decrease the importance in majority class. Hybrid approaches [3], [47], [48] combine data-level methods with algorithm-level methods to better performance. Although these methods can alter the data distribution or modify the penalty matrix, the performance of recognition rate is still limited. Data augmentation of minority classes is similar to the raw data and limits the data diversity. Data augmentation also increases the amount of compute resources and is easily affected by noise. In terms of the penalty matrix, a suitable sensitive cost to the dataset and algorithms requires expert knowledge and is almost impossible to be transferred from one task to another. In contrast, our model is designed for deep learning of end-to-end imbalanced data, and it is suited for imbalanced detection as well as classification tasks.

C. Few-Shot Recognition

In recent years, few-shot learning has shown promise for resolving few-shot problems in regression, classification, and reinforcement learning [49]. Few-shot learning methods are capable of learning and adapting from a few samples and avoid overfitting to the test dataset with novel categories. There are essentially two aspects, model and algorithm, in few-shot learning. Few-shot learning models learn the embedding space from task-specific prior knowledge contained in the training set, such as matching networks [50] and prototypical networks [51]. Gradients decent algorithms, such as Meta-LSTM [17] and model-agnostic metalearning (MAML) [49] have shown impressive results on the few-shot learning datasets MiniImageNet [17] and Ominiglot [52]. The main body of learning is based on CNNs [53] or fully connected neural networks, which have performed well in tasks related to compute vision. A flow of few-shot learning methods shows a continual growing recognition rate in the few-shot dataset [54]. However, the aforementioned work has only been conducted under N-way K-shot classification conditions, which means that examples of few-shot training and few-shot testing are equal. Actually, this setting is not a realistic one, since real-world applications will never be as K-shot as the experiments in these articles. Few-shot learning algorithms that can quickly learn a new task from only a few examples are urgently needed.

III. Proposed CI Loss

In this section, the concept of an ID to measure the degree of imbalance of a dataset is introduced, and then, the proposed novel CI loss is presented.

A. Imbalance Degree

Suppose that we have a dataset $\mathcal{D}$ that has C different classes with class i having N_i samples ($i = 1, \ldots, C$). In order to measure the degree of imbalance of a dataset, we follow [1] and introduce parameter ρ, which is defined as follows:

$$\rho = \frac{\max_i \{N_i\}}{\min_i \{N_i\}}. \tag{1}$$

We can see that ρ is the ratio of the number of samples in the majority class, which has the maximum number of samples, to the number of samples in the minority class, which has the minimum number of samples. However, the ratio ρ denotes the ratio of two extreme cases, whereas it ignores the overall class distribution of the dataset. For instance, three different

distributions [1, 5, 5, 5, 5], [1, 2, 3, 4, 5], and [1, 1, 1, 1, 5] have the same ratio ρ, but different entropy.

To address this issue, we use entropy to measure the degree of imbalance of a dataset

$$\lambda = -\frac{1}{C}\sum_{i=1}^{C}\log\frac{N_i}{\max_i\{N_i\}}. \tag{2}$$

As we will discuss later, the parameter λ works in the proposed CI loss as a constraint to avoid the problem of gradient vanishing because of the huge difference between majority and minority classes. In this article, log means $\log_{10}$.

B. Proposed CI Loss

For a classification task of C categories, according to the commonly used CE loss, the loss function is defined as follows:

$$L_{\text{CE}} = -\sum_{i=1}^{C} y_i \log \hat{y}_i \tag{3}$$

where $y_i \in \{0, 1\}$ specifies the ground truth class and $\hat{y}_i \in [0, 1]$ is the model's estimated probability for the class with ground truth i.

As discussed in [15], the CE loss respects two conditions during model learning. First, the samples in the same category should have category distributions with the identical peak position corresponding to the ground truth label. Second, each class corresponds to a different peak position in the category distribution. For instance, at the initial steps during the training procedure, $\hat{y}_i$ is nearly uniform, i.e., $\frac{1}{C}$. As the training process progresses, the total loss becomes smaller and smaller, and $\hat{y}_i$ will approach 1 for the ground truth class. The CE loss minimizes the training error by assuming that individual samples and classes are equally important. To achieve model generalization with discriminative interclass boundary separation, it is necessary to have a training dataset with sufficiently balanced class frequencies. However, for the dataset with high ID, model training with the conventional CE loss may be suboptimal. The model suffers from generalizing inductive decision boundaries biased toward majority classes while suppressing the contribution of minority classes.

To address this problem, we present CI loss, which is defined as follows:

$$L_{\text{CI}} = -\sum_{i=1}^{C} y_i \left(1 - \sin\left(\frac{\pi}{2}\hat{y}_i\right)\right)\cos\left(\frac{\pi}{2}\hat{y}_i\right) f(\lambda, N_i, N_{\max}) \log \hat{y}_i \tag{4}$$

where $f(\lambda, N_i, N_{\max})$ is defined as follows:

$$f(\lambda, N_i, N_{\max}) = \begin{cases} (N_{\max}/N_i)^{0.5}, & \text{if } \lambda \le 1 \\ (N_{\max}/N_i)^{\frac{1}{8}}, & \text{otherwise} \end{cases} \tag{5}$$

where N_i is the number of samples in class i and $N_{\max} = \max_i\{N_i\}$. λ is the ID defined in (2). We also list the focal loss (FL) [13] as follows for comparison:

$$L_{\text{FL}} = -\sum_{i=1}^{C} y_i (1 - \hat{y}_i)^{\gamma} \log \hat{y}_i. \tag{6}$$

In the left chart in Fig. 2, the CI loss in yellow and blue indicates the situation for the maximum number class and small number class, respectively. The maximum number class, in the yellow line, is beneath the CE loss and above the FL ($\gamma = 5$). Intuitively, the modulating factor with sin and cos (the yellow line) is moderating to γ from 0.5 to 5 of the power function in FL, and the loss is depressed if the ID of the dataset is substantial. When it comes to the blue line, if there is a binary-class recognition task, it can be inferred that the ratio of the two classes is 256:1. The smaller class must descend further to make up for its disadvantage of less quantity. To this end, the value of loss in the blue line is always twice that of the loss in the yellow line.

C. Discussion of CI Loss

We have already stated that deep learning methods with CE loss cannot handle the imbalanced datasets due to the frequency of training samples from majority and minority classes in a batch. Our CI loss has taken frequency into consideration and introduced $f(\lambda, N_i, N_{\max})$ to demonstrate the difference in the amount of category data in the loss. Through this design in loss, the gradient descent will operate in the way of our motivation to alleviate the influence to the parameter modification from imbalance. The following proof illustrates that the derivative of loss, with respect to $\hat{y}_i$, is directly proportional to the gradients. Fig. 3 depicts a simple network with m neurons in the input layer, k neurons in the hidden layer, and n neurons in the output layer ($1 < i < m, 1 < r < k, 1 < j < n$). The given training set is $D = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$. To a training sample (x_s, y_s), the probability sequence is $\hat{y}_s = (\hat{y}_1^s, \ldots, \hat{y}_i^s, \ldots, \hat{y}_n^s)$, and to the jth neuron in the output layer whose bias is θ_j, we can obtain that

$$\hat{y}_j^s = f\left(\sum_{r=1}^{k} w_{rj} h_r - \theta_j\right). \tag{7}$$

If we use the CI loss [formulas (7) and (8)] as the loss function, then the loss to the sample (x_s, y_s) can be presented as follows:

$$L_{\text{CI}}^s = -\sum_{j=1}^{C} y_j^s \left(1 - \sin\left(\frac{\pi}{2}\hat{y}_j^s\right)\right)\cos\left(\frac{\pi}{2}\hat{y}_j^s\right) f(\lambda, N_j, N_{\max}) \log \hat{y}_j^s. \tag{8}$$

The weight parameters w_{rj} from hidden layer to output layer will be updated

$$w_{rj} \leftarrow w_{rj} + \Delta w_{rj}. \tag{9}$$

According to the chain rule, we have

$$\frac{\partial L_{\text{CI}}^s}{\partial w_{rj}} = \frac{\partial L_{\text{CI}}^s}{\partial \hat{y}_j^s}\frac{\partial \hat{y}_j^s}{\partial w_{rj}}. \tag{10}$$

Then, there is $\frac{\partial \hat{y_j^s}}{\partial w_{rj}} = f'(\sum_{r=1}^{k} w_{rj} h_r - \theta_j) h_r$; from (7), we have

$$\frac{\partial L_{\text{CI}}^s}{\partial w_{rj}} = \frac{\partial L_{\text{CI}}^s}{\partial \hat{y_j^s}} f'\left(\sum_{r=1}^{k} w_{rj} h_r - \theta_j\right) h_r. \tag{11}$$

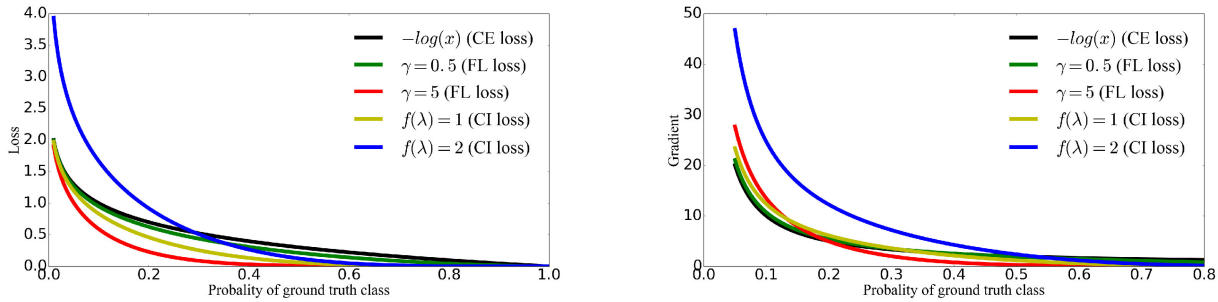


Fig. 2. (Left) Different loss functions in binary classification. (Right) Different gradients of loss. Different colors are related to different loss functions and the gradients. Probability of ground truth class is $\hat{y}_i$. The green line and the red line illustrate the FL in two different parameters (0.5 and 5) of γ. In addition, the black line indicates the CE loss, while the yellow line and the blue line describe the CI loss in different IDs.

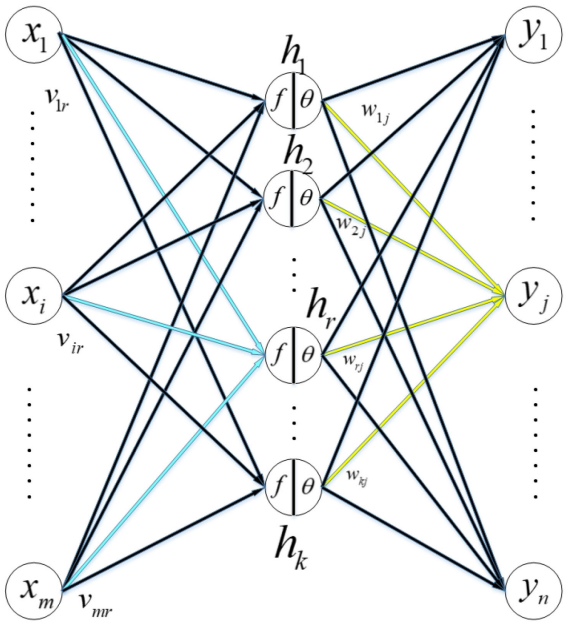


Fig. 3. Multilayer networks.

Generally speaking, the function f is rectified linear unit (ReLU) [55], so the result of f' can only be 0 or 1. Meanwhile, $h_r = (\sum_{i=1}^{k} v_{ir}x_i - \theta_r)$, so it is uncorrelated with $\hat{y}_j^s$. Finally, the second half in (12) can be replaced by constant K:

$$\frac{\partial L_{\mathrm{CI}}^s}{\partial w_{rj}} = \frac{\partial L_{\mathrm{CI}}^s}{\partial \hat{y}_j^s} K. \tag{12}$$

From (12), it can be seen that the relationship between the derivative of loss and gradient can be regarded as directly proportional. Therefore, if we are interested in $\frac{\partial L_{\mathrm{CI}}^s}{\partial w_{rj}}$, it is easier for us to pay attention to $\frac{\partial L_{\mathrm{CI}}^s}{\partial \hat{y}_j^s}$ and overlook the K.

Gradient, affecting the modification of parameters, is influenced by the loss function. It is our wish that gradient descent is able to find the globally optimal solution and avoid the locally optimal solution as much as possible. The derivative of $\hat{y}_i$ from the loss allows us to observe the change of gradient, since the partial derivative is in direct proportion to the gradient. To this end, we differentiate formulas (3), (4), and (7) with respect to $\hat{y}_i$

$$\frac{\partial L_{\mathrm{CE}}}{\partial \hat{y}_i} = -\frac{1}{\ln 10}\frac{y_i}{\hat{y}_i} \tag{13}$$

$$\frac{\partial L_{\mathrm{FL}}}{\partial \hat{y}_i} = -\frac{y_i(1-\hat{y}_i)^{\gamma-1}}{\ln 10}\left(\frac{1}{\hat{y}_i} - \gamma \ln \hat{y}_i - 1\right) \tag{14}$$

$$\frac{\partial L_{\mathrm{CI}}}{\partial \hat{y}_i} = -\frac{y_i f(\lambda)}{\ln 10}\left(\frac{\cos(\frac{\pi}{2}\hat{y}_i)(1-\sin(\frac{\pi}{2}\hat{y}_i))}{\hat{y}_i}\right) + \frac{\pi}{2}\ln \hat{y}_i \left(\cos(\pi \hat{y}_i) + \sin\left(\frac{\pi}{2}\hat{y}_i\right)\right). \tag{15}$$

Because the value of y_i can only be 1 or 0, y_i acts as a switch to let the loss function work merely in the label of such training samples. In the graph on the right in Fig. 2, the gradient of the blue line is much higher than the others when the probability is close to 0, and the step of gradient is two times more than the yellow one. As γ becomes smaller, both the loss and gradient are more approximate to the CE loss. In classification of N categories, $f(\lambda, N_i, N_{\max})$ will contribute to a factor that classes with different number of samples descend the gradient with various speeds, which is determined according to the amount of data. Because the respective factors limit the amount of gradient, the recognition rate of the imbalanced dataset is improved significantly. In summary, categories with a large number of samples, close to the maximum class, are in the normal scale of gradient descent, but the minority classes descend the gradient more according to our function factor $f(\lambda, N_i, N_{\max})$.

However, the factor function $f(\lambda, N_i, N_{\max})$ is not as large as possible. The enormous gradient will bring the algorithm to a situation of gradient vanishing, and loss will not be declined. If parameters are modified thorough enormous gradient, the output will be unstable. Therefore, the best way to train a model is to add the appropriate function factor, rather than a huge function factor, to the loss function, which is the essence of CI loss.

IV. Application Problems

In this section, we apply our proposed CI loss to three object recognition problems: imbalanced image classification, few-shot classification, and object detection.

A. Imbalanced Image Classification

Assume that there are M classes in imbalance training data distribution, which need to be classified by our methods. Then, the notations of training and testing set in two datasets are defined as follows:

$$D_{\text{train}} = \left\{X^1_{\text{train}}, X^2_{\text{train}}, \ldots X^M_{\text{train}}\right\}$$
$$D_{\text{test}} = \left\{X^1_{\text{test}}, X^2_{\text{test}}, \ldots X^M_{\text{test}}\right\}. \tag{16}$$

Each $X^j_* = \{(x^i_{*,j}, y^i_{*,j})\}_{i=1}^{\text{card}(X^j_*)}$, where $*$ is the substitute for the subscript in (16) to distinguish train and test datasets. $x^i_{*,j}$ candidates the ith sample in the jth category in the set $*$. $y^i_{*,j}$ is the label of $x^i_{*,j}$. card(X^j_*) describes the total number of samples in the set X^j_*. Usually, when there are a large number of training samples, people can train a sophisticated classifier $f(x;\theta)$ that inputs unlabeled examples and outputs the correct labels in most cases. They often grasp a classifier perfectly in a nearly uniform dataset, but it is a loss for them to face the imbalance dataset. In deep learning methods, to let the gradient descend more when samples in minority classes occur, we use CI loss to take place of CE loss. This was mentioned in formulas (4)–(8) in Section III-B.

B. Imbalanced Few-Shot Learning

Few-shot classification is a well-established problem in the domain of supervised recognition tasks, where the goal is to learn a classifier to recognize unseen classes when only limited labeled examples are used for training. In the following, we describe the standard formulation of a metalearning-based few-shot classification problem. Specifically, we consider a recent initialization-based method called MAML [49] for few-shot classification.

There is a metalearning dataset D, which contains many classes C. Classes in C are divided into a metatrain set $C_{\text{meta-train}}$ and a metatest set $C_{\text{meta-test}}$ ($C_{\text{meta-train}} \cap C_{\text{meta-test}} = \emptyset$). The number of classes in the metatrain set is much higher than that in the metatest set. If we are conducting N-way K-shot classification, during the process of few-shot training, we sample N classes $(C^i_1, C^i_2, \ldots, C^i_N)$ from the $C_{\text{meta-train}}$ metatrain set (i indicates the ith task). Subsequently, K examples are selected from $C^i_1, C^i_2, \ldots, C^i_N$, and they act as a support set. Meanwhile, other M samples, regarded as a support set, are chosen from the rest of the samples in $C^i_1, C^i_2, \ldots, C^i_N$. M, which can be set to any number, is set to 15 in MAML [49]. These $M \times N$ samples are classified from N-way K-shot, and the gradient of total loss is for updating, according to Algorithm 2 (MAML for Few-Shot Supervised Learning) in [49].

In the metatest process, N classes will be chosen from $C_{\text{meta-test}}$. Then, K-shot is selected from each class as the data to update the model saved from the metatrain process. To evaluate the performance of the model, other K samples are chosen from the rest as in the metatrain process. Results are recorded for every ten or more updates. After repeating the aforementioned process with different sampling hundreds of times, the average recognition rate of the results in each update is indicated as the final behavior of the algorithm.

Nevertheless, in practical applications, it is impossible to find categories with the same number of samples to train or test. In most cases, the examples of classes in a specific task differ, and this can be regarded as an imbalanced dataset. According to the set of metatraining and metatesting, $C^i_1, C^i_2, \ldots, C^i_N$ consist of imbalanced data in both the query set and the support set in metatrain. As shown in Fig. 4, one of the tasks is to learn how to classify the five imbalanced classes of bird, dog, lipstick, fish, and orange. Each update will generate a new modified model, and it will become the input model of the next update. The metatest training update set is composed of imbalanced data, while the rest are equal to evaluate the model's performance. For example, if the task is urged to let the algorithm quickly adapt to the five-way imbalanced dataset with numbers of [1, 2, 3, 4, 5] in each class, the metatest will also be [1, 2, 3, 4, 5] certainly. To satisfy the needs of the test, the data in metatrain should be shaped into the same distribution as the data in metatest. Samples in both the support set and the query set are the distribution of [1, 2, 3, 4, 5], while the query set in metatrain is $[p, 2p, 3p, 4p, 5p]$ (p is an alternative multiple factor and stands for positive integer). However, during the evaluation of the metatest, categories to update and assess are crab, solar panel, stage curtain, towel, and corn, which are totally different from the categories in metatrain, and the number of samples in each class in the query set will be equal ensuring an unbiased recognition rate. Each update also generates an evaluation, from which we will find the performance of the algorithm in imbalanced data. Ordinary metalearning methods for few-shot learning are not fit in the imbalanced datasets, because gradient descent causes a bias from imbalanced data and leads to poor results. However, CI loss plays an important role in adjusting gradient descent, in order to descend gradient more in minority classes. Because of the novel design in CI loss, it can improve the performance of metalearning methods, such as MAML, in an imbalanced few-shot learning dataset.

C. Detection

Object detection has another problem: the large imbalance between the number of labeled object instances and the number of background examples. Although most background examples are easy negatives, this imbalance can make training very inefficient, and the large number of easy negatives tends to overwhelm the training. This problem widely exists in ship detection tasks in optical images. Owing to the special skewing bounding boxes in ships in optical remote sensing images, we have to choose detection networks that can handle skewing labels. We choose R2CNN [18], which is modified from Faster R-CNN [24] and was originally utilized in scene text. What is more, to show the universality of our loss function in ship detection tasks, we have modified the Cascade network [19] from normal bounding boxes to skew bounding boxes.

The training loss of R2CNN is actually a multitask loss, one that includes the loss of axis-aligned boxes, the loss of inclined minimum area boxes, and the loss of classification. The loss

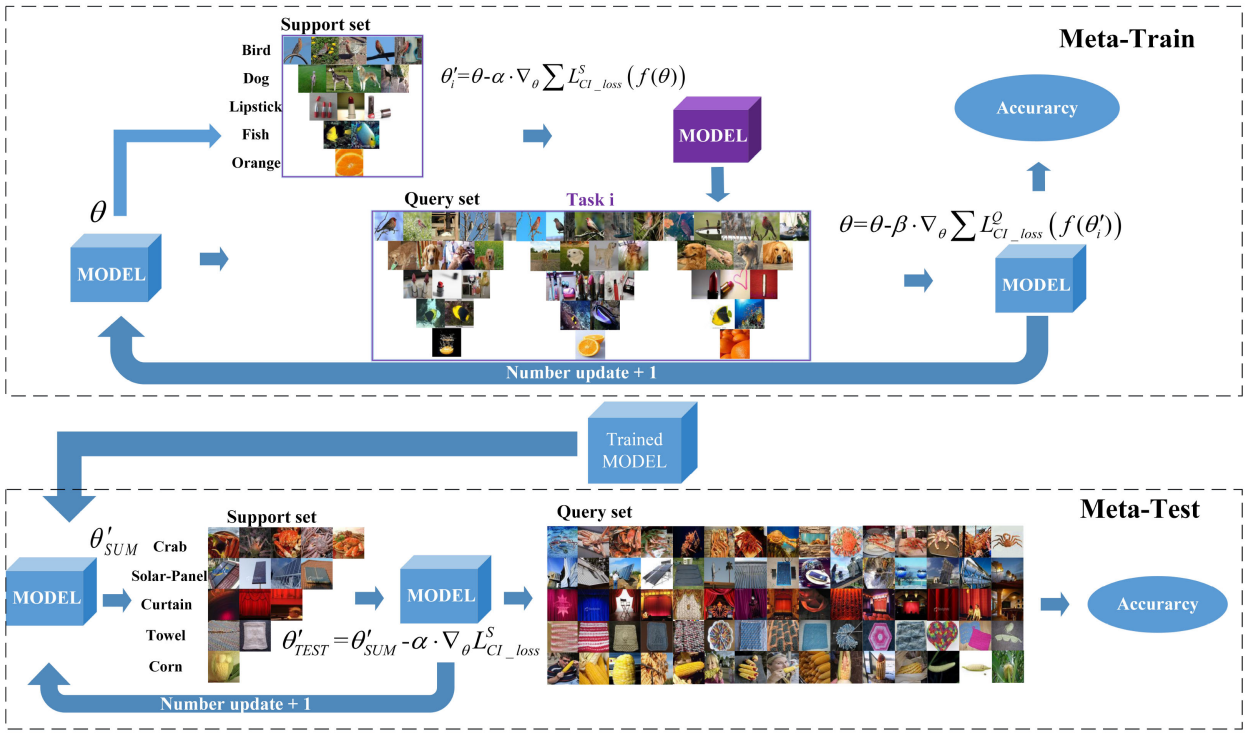


Fig. 4. Process of metatrain and metatest in imbalanced data condition.

function of R2CNN is defined as

$$L(p,t,v,v^*,u,u^*) = L_{\text{cls}}(p,t) + \lambda_1 t\sum_i L_{\text{reg}}(v_i,v_i^*) + \lambda_2 t\sum_i L_{\text{reg}}(u_i,u_i^*). \tag{17}$$

λ_1 and λ_2 are the factors that control the ratio among the three different types of loss. t presents the class label of background and the other different categories of ship. $v = (v_x, v_y, v_w, v_h, v_L)$ is the predicted tuple for the ship label, and $v = (v_x^*, v_y^*, v_w^*, v_h^*, v_L^*)$ presents the ground truth. The information pertaining to vector includes the coordinates of center point, width, height, and label. $u = (u_{x1}, u_{x2}, u_{y1}, u_{y2}, u_h)$ is a tuple that indicates the first two center points of the inclined box and its height, and $u = (u_{x1}^*, u_{x2}^*, u_{y1}^*, u_{y2}^*, u_h^*)$ is the predicted tuple for the ship label. The regression loss is then calculated by smooth L_1 loss [23]. Using k as a substitute for v and u, $L_{\text{reg}}(k_i, k_i^*)$ is defined as follows:

$$L_{\text{reg}}(k,k^*) = \text{smooth}_{L_1}(k-k^*) \tag{18}$$

$$\text{smooth}_{L_1}(x) = \begin{cases} \frac{1}{2}x^2, & \text{if } |x| \le 1 \\ |x| - 0.5, & \text{otherwise} \end{cases}. \tag{19}$$

The classification loss is replaced by our CI loss. p is the prediction output and t is the ground truth label. N_j indicates the total number of the jth category. We record the number of times that samples in the jth category occur in all the training images as N_j. λ is the ID of the ship dataset, taking into consideration the number of samples in every category; then, we have

$$L_{\text{cls}} = -\sum_{j=1}^{N} t\left(1-\sin\left(\frac{\pi}{2}p\right)\right)\cos\left(\frac{\pi}{2}p\right) f(\lambda, N_j, N_{\max})\log p. \tag{20}$$

In the experiments of Cascade networks [19], we modify the original version with skew bounding boxes and change the loss function to CI loss, which is the same as the loss function in (20).

V. Experiments and Results

To verify the effectiveness of our proposed CI loss, extensive experimental evaluations were conducted for the three application tasks. In terms of recognition tasks, our experiments involved classification and detection. The optical ship dataset is imbalanced because of the occurrence of different categories of ships, while the MiniImageNet, MNIST, and MSTAR datasets are sampled to be imbalanced datasets. Experiments were conducted in three aspects:

1) datasets in equal distribution of each class in the training and testing sets for normal sample scale learning (e.g., MNIST and MSTAR);
2) datasets in different distributions of each class in the training and testing sets for few-shot learning (e.g., MiniImageNet);
3) imbalance in both training and testing sets for objection detection and recognition task [e.g., American Optical Ship Recognition (NUDT-AOSR15)].

TABLE I
EVALUATION ON MNIST DATASET

Methods (using stand. split)					Accuracy
Deeply Supervised Nets [[31]]					99.6 %
Generalized Pooling Func. [[32]]					99.7%
Maxout NIN [[33]]					99.8%
Method \ Situation	Stand. split	25% odd digits	25% even digits	10% odd digits	10% even digits
Baseline CNN(CE)	99.3%	98.1%	97.8%	97.6%	97.1%
CoSen CNN	99.3%	98.9%	98.5%	98.6%	98.4%
Baseline CNN(FL)	**99.4%**	98.8%	98.4%	98.3%	98.6%
Baseline CNN(Our CI)	99.3%	**99.1%**	**98.9%**	**98.8%**	**98.7%**

TABLE II
TRAINING AND TESTING SETS IN FOUR SITUATIONS

Situation \ Category	2S1	BMP2	BRDM2	BTR60	BTR70	D7	T62	T72	ZIL131	ZSU234
Train Set(Standard)	299	233	298	256	233	299	299	232	299	299
Test Set(Standard)	274	196	274	195	196	274	273	196	274	274
Situation 1	299	233	298	256	233	24	24	24	24	24
Situation 2	24	24	24	24	24	299	299	232	299	299
Situation 3	299	233	298	256	233	12	12	12	12	12
Situation 4	12	12	12	12	12	299	299	232	299	299

TABLE III
EVALUATION ON THE MSTAR DATASET AND THE BASIC INFORMATION OF FOUR SITUATIONS CAN BE FOUND IN TABLE II

Situation \ Loss	CE	FL	Our CI
Stand. spilit	98.4%	98.3%	**98.5%**
Situation 1	71.2%	72.9%	**75.1%**
Situation 2	70.2%	72.5%	**74.4%**
Situation 3	59.3%	63.1%	**65.1%**
Situation 4	59.5%	63.2%	**64.5%**

All experiments were run on a PC with an Intel single-core i8 CPU, two NVIDIA GTX-1080 Ti GPUs (12-GB VRAM each), and 64-GB RAM. The PC operating system is Ubuntu 18.04. All experiments were carried out using the Python language on the Tensorflow deep learning framework and the CUDA 10.0 toolkit.

A. Experiments on Imbalanced Image Classification

Our results in two imbalanced datasets, MNIST and MSTAR, are shown in Tables I and III. Each CNN method is trained and tested with three different losses. To reduce the possibility of accidental factors of sampling split in both MNIST and MSTAR datasets, we conducted 20 and 100 experiments, including training and testing, respectively, and recorded the average recognition rate in each loss.

In the MNIST dataset, in order to imbalance the categories, we used the same method as Cost-Sensitive [46]. The available training data for the even class were used, and only 10% and 25% of the data in the odd classes were selected as the entire training dataset. Analogously, experiments were also conducted thorough a reversed situation, in which in the even classes, 10% and 25% data were selected, and in the odd classes, the amount of data were normal. The results of four networks are from [31]–[33] and [46], and the CNN in the MNIST dataset has the same configuration as in [46]. In contrast, we only modified the loss and did not use any improvements of augmentation in both the training and testing processes as in [46]. In our results, although the FL showed an elevation on the imbalance dataset, our CI loss outperformed CoSen CNN [46] and CNN with FL or CE loss.

As a matter of fact, the SAR images in the MSTAR dataset were totally different in the imaging mechanism and the pattern (RGB and grayscale) to optical images, which were involved in the MiniImageNet, NUDT-AOSR15, and MNIST datasets. The published MSTAR dataset contains ten types of Soviet military vehicle targets (T-72 and T-62, tanks; BTR-60, BTR-70, BMP-2, and BRDM-2, armored vehicles; ZIL-131, military truck; ZSU-234, self-propelled artillery; 2S1, self-propelled howitzer; and D7, bulldozer), and the specific training and testing samples are shown in Table II. Our experiments were conducted under standard operating conditions; samples in a depression angle of 17° were for training and 15° were for testing. In the training process, we randomly sampled data from five categories in 17°. When not using the procedures of data augmentation, experimental results were below 99.13% with exactly the same CNN structure as in [30]. Because SAR targets are sensitive to azimuth angles, we conducted 100 experiments of random sampling, and the average score is shown in Table III. Substitution in CI loss achieves a performance boost of about 4% and 5% in 24 and 12 to CE loss, respectively.

B. Experiments on Imbalanced Few-Shot Classification

We assessed our method on MiniImageNet, which is a few-shot classification task. There are 64 training classes, 12 validation classes, and 24 test classes, and all images in MiniImageNet were 84×84. This dataset makes it easy to compare our CI

TABLE IV
ASSESSMENT ON IMBALANCED MINIIMAGENET DATASET WITHIN CI LOSS AND CE LOSS

	1,1,1,1,11	1,1,1,2,10	1,1,1,3,9	1,1,1,4,8	1,1,1,5,7	1,1,1,6,6	1,1,2,2,9	1,1,2,3,8	1,1,2,4,7	1,1,2,5,6
$\lambda(ID)$	0.833	0.740	0.668	0.602	0.536	0.466	0.642	0.567	0.495	0.423
MAML(CE)	0.330	0.356	0.350	0.390	0.412	0.414	0.370	0.437	0.445	0.440
MAML(Our CI)	0.402	0.410	0.428	0.470	0.481	0.498	0.456	0.482	0.490	0.504
	1,1,3,3,7	1,1,3,4,6	1,1,3,5,5	1,1,4,4,5	1,2,2,2,8	1,2,2,3,7	1,2,2,4,6	1,2,2,5,5	1,2,3,3,6	1,2,3,4,5
$\lambda(ID)$	0.485	0.407	0.324	0.318	0.541	0.460	0.382	0.299	0.371	0.283
MAML(CE)	0.469	0.482	0.504	0.514	0.448	0.482	0.499	0.516	0.510	0.535
MAML(Our CI)	0.513	0.495	0.539	0.543	0.509	0.519	0.523	0.537	0.525	0.587
	1,2,4,4,4	1,3,3,3,5	1,3,3,4,4	2,2,2,2,7	2,2,2,3,6	2,2,2,4,5	2,2,3,3,5	2,2,3,4,4	2,3,3,3,4	3,3,3,3,3
$\lambda(ID)$	0.180	0.273	0.170	0.435	0.346	0.258	0.248	0.145	0.135	0
MAML(CE)	0.545	0.562	0.571	0.480	0.541	0.573	0.568	0.590	0.591	0.595
MAML(Our CI)	0.560	0.573	0.578	0.518	0.566	0.579	0.582	0.600	0.604	0.605
	1,1,1,1,1	3,3,3,3,3	5,5,5,5,5	1,2,3,4,5	2,4,6,8,10					
$\lambda(ID)$	0	0	0	0.283	0.283					
MAML(CE)	0.481	0.595	0.626	0.535	0.564					
MAML(CI)	0.483	0.605	0.630	0.587	0.601					

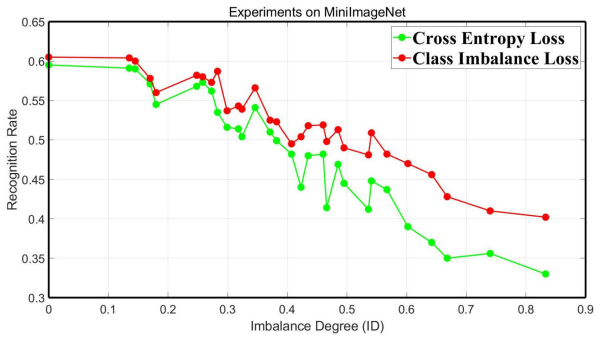


Fig. 5. Experimental results on MiniImageNet.

loss with the CE loss on the basis of MAML. The processes of few-shot training and few-shot testing were introduced in Section IV-A and Fig. 4. We have performed more than 60 experiments on MiniImageNet, with the total sample number of all classes set to 15, and these were divided into five classes with 30 distributions, as shown in (21). For each distribution, we calculated the ID and conducted the experiment with both CI loss and CE loss. The results are presented in Fig. 5 and Table IV

$$A = \left\{ (N_1, N_2, N_3, N_4, N_5) | \sum_{i=1}^{5} N_i = 15 \right\}$$
$$\text{s.t. } 1 \le N_i \le N_j, 1 \le i < j \le 5. \tag{21}$$

The definition of ID is beneficial to sequence datasets, shared same total elements 15 and fixed classes number 5. Next, we analyzed how ID influenced the recognition rate and showed the performance of our loss function. The classifier in few-shot learning is a CNN with four pairs of convolutional layers and max-pooling layers, which act as the feature extractor. The basic information of the CNN is 3×3, stride 1, 64 filters, and ReLU as the activation function. In the end, a fully connected layer works as the distinguishing part. To normalize the output N-dimensional vector of the fully connected layer into the distribution of probability $\hat{p}_t$ for $t = 1, 2, \ldots, N$, we use softmax nonlinearity, as the following formulas show. CE loss is utilized after the softmax layer. p_t is the truth label of N classes. From Table IV, we can see that CI loss is consistently higher than CE loss in every set, which demonstrates the effectiveness of our loss function. The experimental results on imbalanced few-shot image classification (MiniImageNet) are shown in Table IV and Fig. 5. Most of the experiments in Table IV are of five-category classification, and the total number of samples in a set is 15. The green points and red points in Fig. 5 are the results (from Table IV) of CE and CI losses in imbalanced MiniImageNet datasets, respectively. The Y-coordinate indicates the value of recognition rate and the X-coordinate represents the corresponding ID. Fig. 5 shows that when the ID ascends, which means the difference in the amount of data in each category is increased, the recognition rate will decline in both CE loss and CI loss

$$\hat{p}_t = \frac{e^t}{\sum_{i=1}^{N} e^i} \tag{22}$$

$$L_{\text{cls}} = -\sum_{j=1}^{N} p_t \left(1 - \sin\left(\frac{\pi}{2}\hat{p}_t\right)\right) \times \cos\left(\frac{\pi}{2}\hat{p}_t\right) f(\lambda, N_j, N_{\max})) \log \hat{p}_t. \tag{23}$$

C. Experiments on Ship Detection

Our dataset, which is named American Optical Ship Recognition (NUDT-AOSR15), contains 98 images (12 544×12 544 pixels) of different harbors, such as San Diego in the U.S. and Yokosuka in Japan. All images, which were collected from Google Earth, are in 19 levels, and the resolution ratio is 0.6 m per pixel. These remote sensing images are cut into 6000 smaller images, whose pixel size is 1000×1000. Of these, 4800 images are for training and the remaining 1200 images are for testing. There are 15 categories of ships in our dataset, which are shown in Table V. Other types mean that some small ships are

TABLE V
TRAIN AND TEST SETS OF OPTICAL SHIP LOCATION AND RECOGNITION

	Train	Test
(a) Aircraft Carrier (USA)	224	53
(b) Frigate (USA PERRY)	718	182
(c) Cruiser (New Tikang)	683	102
(d) Destroyer(USA BoKe)	1820	451
(e) Frigate (USA Independence)	242	55
(f) Frigate (USA Freedom)	126	30
(g) Amphibious Assault Ship(USA)	283	71
(h) Tanker	51	9
(i) Container Ship	93	21
(j) Grocery Ship	317	79
(k) Amphibious Transport Ship(USA)	787	185
(l) Small Military Warship(USA)	755	205
(m) Supply Ship	423	102
(n) Submarine	2306	541
(o) Others	639	174

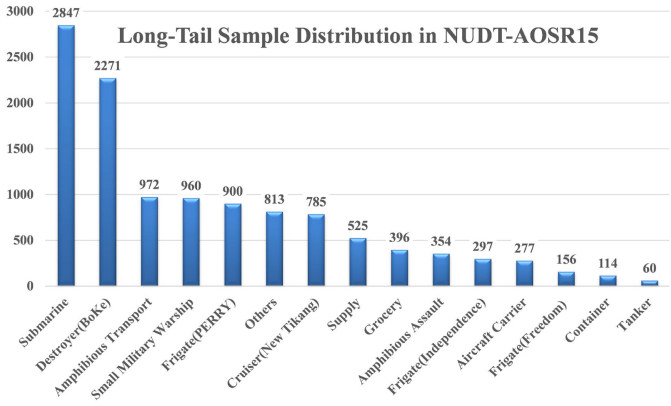


Fig. 6. Long-tail distribution data is one of the features in imbalanced datasets. Our detection and recognition dataset is named American Optical Ship Recognition (NUDT-AOSR15). The amount of data in this bar chart contain both training and testing sets.

inappropriately classified to the other 14 categories. The number of samples in the training set is shown in Fig. 6. From Fig. 7, it is intuitive to find that this is an intractable task for location and recognition. The ship examples in Fig. 7 are the actual size from Google Earth. Ship sizes differ greatly; for example, an aircraft carrier is 30 times larger than a small military warship. Specific information about the training and testing datasets is shown in Table V. There are 2306 samples for submarines, which is more than 45 times the number of samples for tanks.

To evaluate the effectiveness of our loss function, we use the following formulas:

$$\begin{cases} \text{Precision} = \frac{\text{TP}}{\text{TP+FP}} \\ \text{Recall} = \frac{\text{TP}}{\text{TP+FN}} \\ \text{F1}_{\text{score}} = 2\frac{\text{Precision}\times\text{Recall}}{\text{Precision}+\text{Recall}} \\ \text{G}_{\text{mean}} = \sqrt{\text{PrecisionRecall}} \end{cases}. \tag{24}$$

When the value of intersection over union, between the prediction bounding box and the ground truth bounding box, is larger than 0.6 and the label of ship is correct, we observe that such a

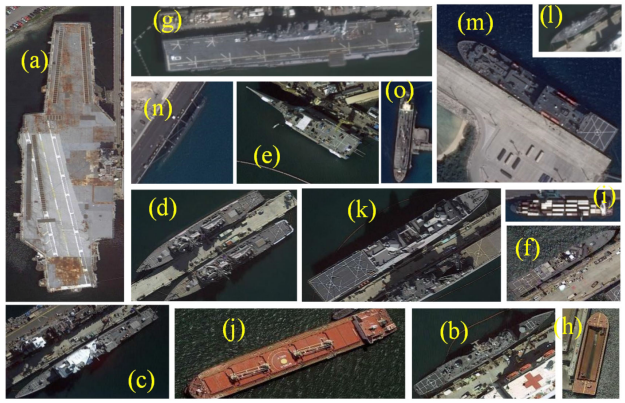


Fig. 7. Ship samples from 15 categories. (NUDT-AOSR15: (a) Aircraft Carrier (USA), (b) Frigate (USA PERRY), (c) Cruiser (New Tikang), (d) Destroyer (USA BoKe), (e) Frigate (USA Independence), (f) Frigate (USA Freedom), (g) Amphibious Assault Ship (USA), (h) Tanker, (i) Container Ship, (j) Grocery Ship, (k) Amphibious Transport Ship (USA), (l) Small Military Warship (USA), (m) Supply Ship, (n) Submarine, and (o) Others.)

prediction bounding box is valid and correct. TP in (24) indicates the total number of correct prediction bounding boxes, while FP represents the bounding boxes that are incorrectly labeled or that have a pure background. FN is the number of overall missing targets in the ground truth bounding boxes. Average precision (AP) is also used as the assessment criterion. When it comes to a given task and category, the precision and recall curve is calculated from a method's ranked output. Recall is defined as the ratio of all positive examples above a given rank, while precision is the proportion of all examples above that rank that are from the positive class. As shown in the following formula, AP summarizes the configuration of the precision/recall curve and is defined as the AP at a set of 11 equally spaced recall levels [0,0.1,...,1]:

$$\text{AP} = \frac{1}{11}\sum_{r\in 0,0.1,\ldots,1} p_{\text{interp}}(r). \tag{25}$$

The precision at each recall level r is interpolated by taking the maximum precision measured for a method for which the related recall exceeds r

$$p_{\text{interp}}(r) = \max_{\widetilde{r}:\widetilde{r}\geq r} p(\widetilde{r}) \tag{26}$$

where $p(\widetilde{r})$ is the measured precision at recall $\widetilde{r}$ [56].

The CNN for feature extraction is based on Resnet [57]. To verify the effectiveness of our loss function, we ran experiments on R2CNN in both Resnet-101 and Resnet-50 with CE loss, FL, and CI loss. The Cascade network in Resnet-101 was modified in a skew bounding box so that it can be used in optical ship detection. The results of optical ship detection are shown in Table VI. Using R2CNN in Resnet-101 as the feature extractor, it is evident that values of AP in Container Ship (i), Amphibious Assault Ship (g), Amphibious Transport Ship (g), and Small Military Warship (l) have been promoted over eight points, and there are about four points of elevation in the PERRY (b), New TiKang (c), Boke (d), Tank (h), Submarine (n), and Others (o) categories by using our CI loss, while only the AP

TABLE VI
RESULTS OF SHIP OBJECT DETECTION IN STANDARD NUDT-AOSR15

			(a)	(b)	(c)	(d)	(e)	(f)	(g)	(h)	(i)	(j)	(k)	(l)	(m)	(n)	(o)	mAP
R2CNN+ Resnet101	CE	Re	1.0	0.95	0.96	0.98	0.98	1.0	0.93	1.0	0.95	0.95	0.99	0.92	0.91	0.93	0.85	
		Pr	0.84	0.80	0.79	0.78	0.82	0.96	0.78	0.18	0.77	0.42	0.62	0.46	0.87	0.77	0.46	
		AP	0.99	0.89	0.91	0.91	0.92	1.0	0.89	0.55	0.85	0.85	0.87	0.60	0.86	0.86	0.69	79.1%
		F1	0.91	0.87	0.87	0.87	0.89	0.98	0.85	0.31	0.85	0.58	0.76	0.61	0.89	0.84	0.60	
		G	0.92	0.87	0.87	0.87	0.9	0.98	0.85	0.42	0.86	0.63	0.78	0.65	0.89	0.85	0.63	
	FL	Re	1.0	0.94	0.96	0.98	0.98	1.0	0.94	1.0	0.95	0.96	0.99	0.93	0.90	0.93	0.84	
		Pr	0.85	0.80	0.78	0.79	0.82	0.97	0.80	0.18	0.83	0.44	0.65	0.45	0.86	0.77	0.47	
		AP	0.99	0.88	0.89	0.91	0.92	1.0	0.90	0.54	0.85	0.87	0.87	0.58	0.85	0.87	0.70	78.9%
		F1	0.92	0.86	0.86	0.87	0.89	0.98	0.86	0.31	0.89	0.60	0.78	0.61	0.88	0.84	0.60	
		G	0.92	0.87	0.87	0.88	0.9	0.98	0.87	0.42	0.89	0.65	0.8	0.65	0.88	0.85	0.63	
	CI	Re	0.98	0.96	0.99	0.98	0.96	1.0	1.0	1.0	0.95	0.96	1.0	0.91	0.88	0.94	0.85	
		Pr	0.83	0.83	0.80	0.82	0.80	1.0	0.83	0.23	0.83	0.49	0.69	0.51	0.88	0.81	0.55	
		AP	0.96	0.93	0.94	0.94	0.92	1.0	0.98	0.59	0.94	0.90	0.95	0.69	0.86	0.90	0.75	**82.9%**
		F1	0.9	0.89	0.88	0.89	0.87	1.0	0.91	0.37	0.89	0.65	0.82	0.65	0.88	0.87	0.67	
		G	0.9	0.89	0.89	0.9	0.88	1.0	0.91	0.48	0.89	0.69	0.83	0.68	0.88	0.87	0.68	
R2CNN+ Resnet50	CE	Re	0.96	0.86	0.91	0.97	0.98	1.0	0.99	0.89	0.90	0.83	0.98	0.74	0.81	0.87	0.71	
		Pr	0.88	0.79	0.76	0.74	0.78	0.88	0.70	0.17	0.56	0.42	0.66	0.57	0.80	0.78	0.55	
		AP	0.95	0.76	0.82	0.85	0.87	1.0	0.91	0.65	0.80	0.66	0.77	0.57	0.73	0.79	0.58	73.2%
		F1	0.92	0.82	0.83	0.84	0.87	0.94	0.82	0.29	0.69	0.56	0.79	0.64	0.8	0.82	0.62	
		G	0.92	0.82	0.83	0.85	0.87	0.94	0.83	0.39	0.71	0.59	0.8	0.65	0.8	0.82	0.62	
	FL	Re	0.94	0.87	0.91	0.95	0.98	1.0	0.97	0.89	0.90	0.78	0.97	0.76	0.81	0.89	0.73	
		Pr	0.88	0.80	0.76	0.74	0.78	0.97	0.73	0.18	0.58	0.40	0.73	0.57	0.83	0.78	0.53	
		AP	0.93	0.76	0.83	0.84	0.88	1.0	0.89	0.64	0.86	0.64	0.89	0.58	0.76	0.81	0.61	73.9%
		F1	0.91	0.83	0.83	0.83	0.87	0.98	0.83	0.3	0.71	0.53	0.83	0.65	0.82	0.83	0.61	
		G	0.91	0.83	0.83	0.84	0.87	0.98	0.84	0.4	0.72	0.56	0.84	0.66	0.82	0.83	0.62	
	CI	Re	0.92	0.88	0.95	0.98	0.96	1.0	0.94	0.89	0.95	0.77	0.99	0.69	0.83	0.91	0.74	
		Pr	0.84	0.84	0.79	0.79	0.79	0.91	0.76	0.18	0.83	0.47	0.70	0.65	0.81	0.80	0.65	
		AP	0.92	0.89	0.89	0.92	0.89	1.0	0.92	0.71	0.94	0.67	0.92	0.57	0.75	0.84	0.68	**77.0%**
		F1	0.88	0.86	0.86	0.87	0.87	0.95	0.84	0.3	0.89	0.58	0.82	0.67	0.82	0.85	0.69	
		G	0.88	0.86	0.87	0.88	0.87	0.95	0.85	0.4	0.89	0.6	0.83	0.67	0.82	0.85	0.69	
Cascade+ Resnet101	CE	Re	0.94	0.93	0.92	0.96	0.98	1.0	0.99	0.89	0.81	0.81	0.98	0.76	0.92	0.78	0.77	
		Pr	0.96	0.96	0.94	0.97	0.94	0.96	1.0	0.73	0.74	0.89	0.94	0.80	0.95	0.79	0.87	
		AP	0.94	0.92	0.91	0.96	0.96	1.0	0.98	0.81	0.79	0.76	0.97	0.68	0.92	0.71	0.75	81.6%
		F1	0.95	0.94	0.93	0.96	0.96	0.98	0.99	0.8	0.77	0.85	0.96	0.78	0.93	0.78	0.82	
		G	0.95	0.94	0.93	0.96	0.96	0.98	0.99	0.81	0.77	0.85	0.96	0.78	0.93	0.78	0.82	
	FL	Re	0.96	0.92	0.92	0.97	0.98	1.0	0.98	1.0	0.81	0.98	0.97	0.76	0.92	0.79	0.76	
		Pr	1.0	0.95	0.95	0.97	0.95	0.97	0.98	0.81	0.80	0.88	0.93	0.80	0.94	0.80	0.86	
		AP	0.96	0.90	0.91	0.96	0.96	1.0	0.97	0.97	0.79	0.76	0.96	0.67	0.93	0.72	0.73	82.4%
		F1	0.98	0.93	0.93	0.97	0.96	0.98	0.98	0.9	0.8	0.93	0.95	0.78	0.93	0.79	0.81	
		G	0.98	0.93	0.93	0.97	0.96	0.98	0.98	0.9	0.8	0.93	0.95	0.78	0.93	0.79	0.81	
	CI	Re	0.94	0.92	0.94	0.97	0.98	1.0	0.99	1.0	0.81	0.85	0.98	0.77	0.93	0.80	0.77	
		Pr	0.98	0.95	0.95	0.97	0.95	0.97	1.0	0.82	0.83	0.90	0.94	0.80	0.94	0.82	0.87	
		AP	0.94	0.90	0.93	0.97	0.97	1.0	0.99	0.97	0.80	0.80	0.97	0.69	0.93	0.75	0.75	**83.4%**
		F1	0.96	0.93	0.94	0.97	0.96	0.98	0.99	0.9	0.82	0.87	0.96	0.78	0.93	0.81	0.82	
		G	0.96	0.93	0.94	0.97	0.96	0.98	0.99	0.91	0.82	0.87	0.96	0.78	0.93	0.81	0.82	

value of Aircraft Carrier (a) shows a decline of 3%. Because some types of Amphibious Assault Ship (g) are similar to the Aircraft Carrier (a) [for instance, the LHA-1 Tarawa class and Wasp-class amphibious assault ship closely resemble Aircraft Carrier (a)], we see an 8% increase in Amphibious Assault Ship (g) through CI loss, which contributes to a slight decrease in Aircraft Carrier (a). Some Aircraft Carriers (a) are classified as Amphibious Assault Ships (g) by mistake. The mean AP in R2CNN in Resnet-101 occupies 82.9%, which exceeds 3.8% and 4% over CE loss and FL, respectively. The Cascade network has a higher mean average precision (mAP) in CI loss than CE loss and FL as well. Table VII shows six situations whose train sets are sampled randomly from NUDT-AOSR15, and whose test sets are the same as in Table VI. The results of these six situations with compared losses show the effectiveness of our CI loss. For the sake of brevity, G and F1 scores are omitted in Table VII, and only AP and mAP are shown.

D. Results Discussion

In this section, we provide discuss and analysis of the experiment results obtained by our proposed CI loss in different application problems.

1) Results for Imbalanced Image Classification: Results on the MNIST and MSTAR datasets are shown in Tables I and III. Because the MNIST dataset is easy for algorithm to recognize, even in the imbalanced situations, different methods still achieve a high performance. The results of baseline CNN with CE loss have already achieved 97%; thus, it is difficult to elevate the accuracy even a bit. It is obvious that all the improved methods overcome the CE loss in four imbalanced situations. Our CI

TABLE VII
RESULTS OF SHIP OBJECT DETECTION IN SAMPLED NUDT-AOSR15 IN R2CNN WITH RESNET101

			(a)	(b)	(c)	(d)	(e)	(f)	(g)	(h)	(i)	(j)	(k)	(l)	(m)	(n)	(o)	mAP
	Num		38	457	398	607	190	63	189	44	81	276	515	608	304	1841	524	
		Re	0.91	0.83	0.90	0.96	0.84	1.00	0.93	1.00	0.76	0.77	0.97	0.83	0.67	0.88	0.76	
	CE	Pr	0.73	0.72	0.73	0.75	0.74	0.81	0.69	0.14	0.50	0.33	0.61	0.43	0.85	0.72	0.39	
		AP	0.90	0.74	0.80	0.84	0.69	0.99	0.82	0.37	0.62	0.62	0.76	0.51	0.60	0.79	0.57	66.02%
Sit. 1		Re	0.91	0.80	0.91	0.96	0.84	1.00	0.92	1.00	0.76	0.78	0.96	0.83	0.70	0.88	0.75	
	FL	Pr	0.73	0.71	0.74	0.75	0.75	0.83	0.66	0.15	0.50	0.31	0.61	0.42	0.87	0.72	0.39	
		AP	0.90	0.69	0.82	0.83	0.69	0.99	0.81	0.41	0.64	0.63	0.74	0.52	0.64	0.78	0.57	66.52%
		Re	0.91	0.85	0.92	0.97	0.88	1	0.92	1	0.77	0.79	0.96	0.84	0.7	0.90	0.79	
	CI	Pr	0.74	0.73	0.74	0.74	0.76	0.81	0.67	0.20	0.52	0.32	0.60	0.42	0.87	0.72	0.45	
		AP	0.90	0.76	0.82	0.84	0.72	1	0.82	0.43	0.68	0.62	0.74	0.53	0.63	0.80	0.60	**68.15%**
	Num		230	552	114	607	121	88	207	51	93	213	556	672	349	2151	610	
		Re	0.85	0.40	0.89	0.91	0.98	0.57	0.92	1.00	0.76	0.86	0.98	0.85	0.66	0.90	0.77	
	CE	Pr	0.78	0.30	0.68	0.75	0.74	0.94	0.71	0.23	0.42	0.24	0.61	0.38	0.87	0.60	0.35	
		AP	0.82	0.39	0.77	0.77	0.86	0.57	0.81	0.32	0.70	0.71	0.77	0.52	0.61	0.78	0.39	63.01%
Sit. 2		Re	0.85	0.45	0.91	0.91	0.98	0.83	0.92	1.00	0.76	0.87	1.00	0.87	0.64	0.89	0.78	
	FL	Pr	0.78	0.32	0.73	0.77	0.76	0.89	0.72	0.23	0.44	0.23	0.62	0.37	0.86	0.58	0.35	
		AP	0.82	0.40	0.81	0.82	0.88	0.81	0.82	0.31	0.71	0.70	0.83	0.52	0.57	0.76	0.40	64.38%
		Re	0.85	0.79	0.94	0.93	0.98	0.83	0.95	1.00	0.76	0.87	1.00	0.87	0.64	0.89	0.78	
	CI	Pr	0.83	0.72	0.75	0.75	0.82	0.90	0.72	0.23	0.54	0.23	0.62	0.37	0.86	0.58	0.36	
		AP	0.85	0.69	0.83	0.82	0.91	0.81	0.83	0.32	0.75	0.71	0.83	0.52	0.58	0.77	0.40	**66.54%**
	Num		113	359	352	1003	129	67	142	32	52	208	434	449	240	1351	373	
		Re	0.91	0.74	0.04	0.95	0.78	0.77	0.87	1.00	0.81	0.78	0.95	0.77	0.73	0.87	0.66	
	CE	Pr	0.84	0.70	0.90	0.61	0.81	0.79	0.78	0.13	0.65	0.36	0.63	0.40	0.89	0.72	0.41	
		AP	0.90	0.60	0.04	0.78	0.72	0.71	0.80	0.74	0.77	0.67	0.70	0.52	0.68	0.77	0.36	61.06%
Sit. 3		Re	0.96	0.80	0.32	0.95	0.78	0.83	0.92	1.00	0.81	0.84	0.97	0.79	0.66	0.90	0.74	
	FL	Pr	0.84	0.62	0.88	0.71	0.81	0.64	0.76	0.13	0.65	0.38	0.65	0.40	0.91	0.71	0.41	
		AP	0.96	0.64	0.29	0.80	0.73	0.70	0.85	0.62	0.79	0.71	0.77	0.53	0.62	0.81	0.45	63.14%
		Re	0.96	0.81	0.32	0.94	0.82	0.80	0.92	1.00	0.81	0.85	0.97	0.83	0.74	0.89	0.75	
	CI	Pr	0.86	0.65	0.90	0.65	0.75	0.75	0.77	0.20	0.65	0.35	0.69	0.39	0.86	0.69	0.39	
		AP	0.97	0.66	0.30	0.80	0.76	0.70	0.86	0.69	0.79	0.71	0.78	0.54	0.68	0.79	0.44	**66.25%**
	Num		151	386	342	1003	129	63	142	32	53	208	440	457	242	1352	377	
		Re	0.94	0.01	0.12	0.95	0.60	0.77	0.89	1.00	0.76	0.87	0.98	0.80	0.50	0.88	0.77	
	CE	Pr	0.81	0.67	0.83	0.61	0.83	0.61	0.63	0.13	0.67	0.34	0.57	0.40	0.86	0.69	0.36	
		AP	0.93	0.01	0.11	0.75	0.51	0.70	0.73	0.33	0.70	0.77	0.74	0.49	0.46	0.78	0.41	52.10%
Sit. 4		Re	0.94	0.02	0.13	0.95	0.65	0.80	0.87	1.00	0.76	0.89	0.98	0.81	0.52	0.89	0.75	
	FL	Pr	0.76	0.60	0.84	0.59	0.80	0.62	0.61	0.11	0.57	0.26	0.53	0.37	0.87	0.66	0.29	
		AP	0.92	0.01	0.12	0.73	0.55	0.75	0.73	0.40	0.69	0.75	0.75	0.47	0.47	0.77	0.37	53.01%
		Re	0.94	0.06	0.20	0.94	0.65	0.77	0.90	1.00	0.76	0.89	0.98	0.81	0.51	0.90	0.80	
	CI	Pr	0.77	0.83	0.87	0.62	0.80	0.64	0.63	0.18	0.62	0.26	0.53	0.37	0.84	0.66	0.29	
		AP	0.93	0.05	0.19	0.72	0.55	0.70	0.75	0.42	0.71	0.76	0.75	0.48	0.46	0.78	0.39	**54.45%**
	Num		21	132	438	662	66	71	176	24	51	310	502	629	308	2110	582	
		Re	0.60	0.06	0.93	0.43	0.20	0.87	0.87	0.67	0.71	0.89	0.97	0.84	0.60	0.88	0.66	
	CE	Pr	0.84	0.61	0.37	0.82	0.85	0.79	0.56	0.14	0.65	0.52	0.60	0.45	0.87	0.70	0.32	
		AP	0.60	0.04	0.77	0.36	0.20	0.86	0.75	0.20	0.68	0.82	0.72	0.57	0.54	0.78	0.30	51.18%
Sit. 5		Re	0.60	0.38	0.92	0.85	0.31	0.77	0.92	1.00	0.71	0.86	0.97	0.84	0.61	0.89	0.68	
	FL	Pr	0.74	0.73	0.55	0.74	0.77	0.82	0.59	0.16	0.65	0.55	0.59	0.45	0.87	0.73	0.34	
		AP	0.60	0.28	0.77	0.67	0.29	0.74	0.80	0.28	0.69	0.80	0.74	0.56	0.55	0.79	0.31	55.46%
		Re	0.62	0.42	0.94	0.86	0.38	0.87	0.93	1.00	0.71	0.86	0.98	0.85	0.66	0.90	0.69	
	CI	Pr	0.75	0.68	0.58	0.74	0.78	0.85	0.61	0.15	0.65	0.55	0.59	0.45	0.86	0.74	0.31	
		AP	0.61	0.31	0.81	0.69	0.36	0.85	0.80	0.25	0.69	0.79	0.76	0.56	0.59	0.81	0.32	**57.72%**
	Num		184	66	125	499	88	58	155	51	86	307	500	606	293	2145	595	
		Re	0.94	0.01	0.16	0.94	0.60	0.80	0.89	1.00	0.76	0.90	0.98	0.81	0.53	0.88	0.76	
	CE	Pr	0.76	1.00	0.84	0.60	0.83	0.62	0.59	0.10	0.57	0.27	0.53	0.36	0.86	0.64	0.29	
		AP	0.92	0.01	0.15	0.72	0.51	0.77	0.74	0.34	0.70	0.76	0.73	0.47	0.48	0.76	0.36	52.61%
Sit. 6		Re	0.94	0.18	0.44	0.93	0.67	0.80	0.92	1.00	0.76	0.89	0.96	0.84	0.58	0.89	0.79	
	FL	Pr	0.76	0.78	0.76	0.63	0.80	0.73	0.62	0.11	0.62	0.25	0.53	0.37	0.91	0.66	0.30	
		AP	0.93	0.15	0.39	0.73	0.56	0.73	0.75	0.38	0.73	0.75	0.75	0.47	0.55	0.77	0.37	56.35%
		Re	0.94	0.18	0.45	0.93	0.67	0.80	0.92	1.00	0.76	0.89	0.96	0.84	0.58	0.90	0.80	
	CI	Pr	0.74	0.78	0.77	0.63	0.80	0.73	0.62	0.11	0.59	0.25	0.53	0.40	0.91	0.68	0.32	
		AP	0.93	0.15	0.40	0.74	0.56	0.73	0.75	0.62	0.72	0.74	0.74	0.48	0.55	0.78	0.38	**58.85%**

"Sit." means situation.

loss slightly exceeds FL and CoSen CNN, while CoSen loss is only able to be used in classification tasks. When it comes to the MSTAR dataset, the results in imbalanced situations witness sharp drops in all the experiments. However, the results in CI loss are about 4% and 2% higher than those in CE loss and FL, respectively. This indicates that our proposed CI loss is more effective than FL. These results in two imbalanced image classification datasets, including optical and SAR images, have proved that our CI loss indeed overcomes the imbalance in some extent and promotes the performance.

2) Results for Imbalanced Few-Shot Image Classification: From experiments on imbalanced few-shot classification in Table IV and Fig. 5, the bigger the ID is, the more the CI loss exceeds CE loss in recognition rate. The point on the red line in Fig. 5, indicating CI loss, is approximately 0.4 in the extreme situation [1,1,1,1,11], while the point on the green line only occupies 0.325. The recognition rate to the ID is in the monotonic decreasing tendency, while, in some points, there are some fluctuations. This is because the categories are randomly selected from the whole training and testing sets. In the metatest process, there are only 600 random experiments. Thus, it is impossible to traverse all of the categories and images.

From the above conclusion, it is obvious that if the class number and the total number of samples are fixed, the more imbalanced dataset will cause a lower recognition rate. Furthermore, in the condition of the same ID, the more the number of data samples, the higher the recognition rate. For instance, data amount distribution of [1,1,1,1,1], [3,3,3,3,3], and [5,5,5,5,5] is the same as in ID (ID $= 0$), but the recognition rates are 0.481, 0.595, and 0.626 in CE loss, separately. In addition, the results of CE loss and CI loss are very close. When the ID is not 0, just like data distribution [1,2,3,4,5] and [2,4,6,8,10], the recognition rates are 0.535 & 0.587 and 0.564 & 0.601 in CE loss and CI loss, respectively.

3) Results for Ship Detection and Recognition: We conduct abundant experiments on NUDT-AOSR15 with different detection methods (R2CNN and Cascade), different convolution neural networks (Resnet50 and Resnet101), and various imbalanced situations. In terms of the results in standard NUDT-AOSR15 in Table VI, Cascade with Resnet101 and CI loss achieves the highest mAP, 83.4%, among other compared experiments. The results of FL are approximate to the results of CE loss, which is nearly 4% lower than our CI loss. Furthermore, the AP of four to five categories in three sets of experiments is improved significantly in the results of our CI loss to CE loss, which contributes to the increase in the mAP.

Except conducting the experiments on the standard NUDT-AOSR15, we have also conducted enough number of experiments on the random sampled NUDT-AOSR15. The number of samples in each category and the relevant results are shown in Table VII. We train the detection network (R2CNN with ResNet101) only with different number of losses in each situation. Owing to lower data amount than standard NUDT-AOSR15, all the results of experiments undergo a decrease in mAP. Overall, FL is able to ameliorate the imbalance in the datasets, but CI loss still get the highest mAP in all the experiments. Almost all the APs in every categories have been elevated through CI loss. AP values of some categories increase impressively, and this will also bring several mistakes in the testing process. Thus, AP values of one or two categories in CI loss are a bit lower than those in CE loss. The mAP values in six situations with different training samples and different amount of data of training samples, which are randomly sampled from standard NUDT-AOSR15, demonstrate the effectiveness of our proposed CI loss. This means that our proposed CI loss can be used as one of the powerful methods to overcome the imbalance of datasets.

To sum up, the imbalance of the dataset will influence the performance of algorithms in recognition tasks. In addition, the higher the ID, the poorer the results of algorithms. When methods that are proposed to solve with the class imbalance problem are used, the recognition rate or mAP will increase. Our proposed CI loss is competitive among the state-of-the-art methods.

VI. Conclusion

In this article, we realize that class imbalance is the general problem in tasks, which chiefly come down to classification. In the problems of image recognition and multicategory detection, CI influences the gradient descent so that the trained model performs well on majority classes, but poorly on minority classes. CE loss is usable in class imbalance tasks. To address this, we propose CI loss to handle the class imbalance problems in classification and detection datasets. The essence of CI loss is to descend the gradient more in the training process, by taking the probability of occurrence in different classes into consideration. In order to limit the ratio between classes, ID is also proposed as a factor in our loss function and acts as a reasonable assessment criterion in arbitrary datasets. Experimental results and their analysis demonstrate that our loss function achieves higher accuracy than other methods or losses. We hope to foster more progress in the improvement of loss functions.

References

[1] J. M. Johnson and T. M. Khoshgoftaar, "Survey on deep learning with class imbalance," *J. Big Data*, vol. 6, no. 1, 2019, Art. no. 27.
[2] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 248–255.
[3] H. He and E. Garcia, "Learning from imbalanced data," *IEEE Trans. Knowl. Data Eng.*, vol. 21, no. 9, pp. 1263–1284, Sep. 2009.
[4] W. Wei, J. Li, L. Cao, Y. Ou, and J. Chen, "Effective detection of sophisticated online banking fraud on extremely imbalanced data," *World Wide Web*, vol. 16, no. 4, pp. 449–475, 2013.
[5] T. Berg *et al.*, "Birdsnap: Large-scale fine-grained visual categorization of birds," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 2011–2018.
[6] Y. Zhong *et al.*, "Unequal-training for deep face recognition with long-tailed noisy data," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 7812–7821.
[7] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, 2017.
[8] L. Liu *et al.*, "Deep learning for generic object detection: A survey," *Int. J. Comput. Vis.*, vol. 128, pp. 261–318, 2020.
[9] Y. Lecun, Y. Bengio, and G. E. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.
[10] O. Russakovsky *et al.*, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, 2015.

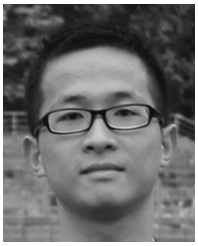
[11] Z. Liu, Z. Miao, X. Zhan, J. Wang, B. Gong, and S. X. Yu, "Large-scale long-tailed recognition in an open world," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 2537–2546.
[12] C. Huang, Y. Li, C. L. Chen, and X. Tang, "Deep imbalanced learning for face recognition and attribute prediction," *IEEE Trans. Pattern Anal. Mach. Intell.*, to be published.
[13] T. Lin, P. Goyal, R. B. Girshick, K. He, and P. Dollar, "Focal loss for dense object detection," in *Proc. Int. Conf. Comput. Vis.*, 2017, pp. 2999–3007.
[14] F. Wu, X.-Y. Jing, S. Shan, W. Zuo, and J.-Y. Yang, "Multiset feature learning for highly imbalanced data classification," in *Proc. 31st AAAI Conf. Artif. Intell.*, 2017, pp. 1583–1589.
[15] Q. Dong, S. Gong, and X. Zhu, "Imbalanced deep learning by minority class incremental rectification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 6, pp. 1367–1381, Jun. 2019.
[16] A. Fernandez, S. Garcia, M. J. D. Jesus, and F. Herrera, "A study of the behaviour of linguistic fuzzy rule based classification systems in the framework of imbalanced data-sets," *Fuzzy Sets Syst.*, vol. 159, no. 18, pp. 2378–2398, 2008.
[17] S. Ravi and H. Larochelle, "Optimization as a model for few-shot learning," in *Proc. 5th Int. Conf. Learn. Representations (ICLR)*, Toulon, France, Apr. 24–26, 2017.
[18] Y. Jiang *et al.*, "R2CNN: Rotational region CNN for orientation robust scene text detection," 2017, *arXiv:1706.09579*.
[19] Z. Cai and N. Vasconcelos, "Cascade R-CNN: Delving into high quality object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 6154–6162.
[20] W. Liu *et al.*, "SSD: Single shot multibox detector," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 21–37.
[21] J. Redmon, S. K. Divvala, R. B. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 779–788.
[22] R. B. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 580–587.
[23] R. B. Girshick, "Fast R-CNN," in *Proc. Int. Conf. Comput. Vis.*, 2015, pp. 1440–1448.
[24] S. Ren, K. He, R. B. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.
[25] R. Zhang, J. Yao, K. Zhang, C. Feng, and J. Zhang, "S-CNN-based ship detection from high-resolution remote sensing images," *Int. Arch. Photogrammetry, Remote Sens. Spatial Inf. Sci.*, vol. XLI-B7, pp. 423–430, 2016.
[26] Z. Liu, J. Hu, L. Weng, and Y. Yang, "Rotated region based CNN for ship detection," in *Proc. IEEE Int. Conf. Image Process.*, 2017, pp. 900–904.
[27] S. Nie, Z. Jiang, H. Zhang, B. Cai, and Y. Yao, "Inshore ship detection based on mask R-CNN," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2018, pp. 693–696.
[28] H. Li, Z. Lin, X. Shen, J. Brandt, and G. Hua, "A convolutional neural network cascade for face detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 5325–5334.
[29] Y. LeCun and Y. Bengio, "Convolutional networks for images, speech, and time series," in *The Handbook of Brain Theory and Neural Networks.* Cambridge, MA, USA: MIT Press, 1998.
[30] S. Chen, H. Wang, F. Xu, and Y. Jin, "Target classification using the deep convolutional networks for SAR images," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 8, pp. 4806–4817, Aug. 2016.
[31] C. Lee, S. Xie, P. W. Gallagher, Z. Zhang, and Z. Tu, "Deeply-supervised nets," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2015, pp. 562–570.
[32] C. Lee, P. W. Gallagher, and Z. Tu, "Generalizing pooling functions in convolutional neural networks: Mixed, gated, and tree," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2016, pp. 464–472.
[33] J.-R. Chang and Y.-S. Chen, "Batch-normalized maxout network in network," *CoRR*, 2015. [Online]. Available: http://arxiv.org/abs/1511.02583
[34] J. Park, S. Park, and K. Kim, "New discrimination features for SAR automatic target recognition," *IEEE Geosci. Remote Sens. Lett.*, vol. 10, no. 3, pp. 476–480, May 2013.
[35] Z. Jianxiong, S. Zhiguang, C. Xiao, and F. Qiang, "Automatic target recognition of SAR images based on global scattering center model," *IEEE Trans. Geosci. Remote Sens.*, vol. 49, no. 10, pp. 3713–3729, Oct. 2011.
[36] G. Dong, G. Kuang, N. Wang, L. Zhao, and J. Lu, "SAR target recognition via joint sparse representation of monogenic signal," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 8, no. 7, pp. 3316–3328, Jul. 2015.
[37] G. Dong and G. Kuang, "Target recognition in SAR images via classification on riemannian manifolds," *IEEE Geosci. Remote Sens. Lett.*, vol. 12, no. 1, pp. 199–203, Jan. 2015.
[38] G. Dong, G. Kuang, N. Wang, and W. Wang, "Classification via sparse representation of steerable wavelet frames on Grassmann manifold: Application to target recognition in SAR image," *IEEE Trans. Image Process.*, vol. 26, no. 6, pp. 2892–2904, Jun. 2017.
[39] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," *J. Artif. Intell. Res.*, vol. 16, no. 1, pp. 321–357, 2002.
[40] H. Han, W. Wang, and B. Mao, "Borderline-smote: A new over-sampling method in imbalanced data sets learning," in *Proc. Int. Conf. Intell. Comput.*, 2005, pp. 878–887.
[41] C. Bunkhumpornpat, K. Sinapiromsaran, and C. Lursinsap, "Safe-level-smote: Safe-level-synthetic minority over-sampling technique for handling the class imbalanced problem," in *Proc. Pacific-Asia Conf. Knowl. Discovery Data Mining*, 2009, pp. 475–482.
[42] I. Goodfellow *et al.*, "Generative adversarial nets," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2014, pp. 2672–2680.
[43] S. Benaim and L. Wolf, "One-shot unsupervised cross domain translation," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2018, pp. 2104–2114.
[44] H. Qi, M. S. Brown, and D. G. Lowe, "Low-shot learning with imprinted weights," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 5822–5830.
[45] A. Santoro, S. Bartunov, M. Botvinick, D. Wierstra, and T. Lillicrap, "Meta-learning with memory-augmented neural networks," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 1842–1850.
[46] S. H. Khan, M. Hayat, M. Bennamoun, F. A. Sohel, and R. Togneri, "Cost-sensitive learning of deep feature representations from imbalanced data," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 8, pp. 3573–3587, Aug. 2018.
[47] S. S. Girija, "TensorFlow: Large-scale machine learning on heterogeneous distributed systems," *Software Available From Tensorflow. Org*, vol. 39, 2016.
[48] X.-Y. Liu, J. Wu, and Z.-H. Zhou, "Exploratory undersampling for class-imbalance learning," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 39, no. 2, pp. 539–550, Apr. 2009.
[49] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *Proc. 34th Int. Conf. Mach. Learn.*, 2017, pp. 1126–1135.
[50] O. Vinyals, C. Blundell, T. P. Lillicrap, K. Kavukcuoglu, and D. Wierstra, "Matching networks for one shot learning," in*Proc. 30th Int. Conf. Neural Inf. Process. Syst.*, 2016, pp. 3637–3645.
[51] J. Snell, K. Swersky, and R. S. Zemel, "Prototypical networks for few-shot learning," in *Proc. Int Conf. Neural Inf. Process. Syst.*, 2017, pp. 4077–4087.
[52] B. M. Lake, R. Salakhutdinov, J. Gross, and J. B. Tenenbaum, "One shot learning of simple visual concepts," *Cogn. Sci.*, vol. 33, no. 33, pp. 2568–2573, 2011.
[53] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 1–9.
[54] E. Triantafillou *et al.*, "Meta-dataset: A dataset of datasets for learning to learn from few examples," in *Proc. Int. Conf. Learn. Representations (submission)*, 2020.
[55] V. Nair and G. E. Hinton, "Rectified linear units improve restricted Boltzmann machines," in *Proc. 27th Int. Conf. Mach. Learn.*, 2010, pp. 807–814.
[56] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (VOC) challenge," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, 2010.
[57] 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.

Linbin Zhang received the B.S. degree in information engineering, in 2018 from the National University of Defense Technology, Changsha, China, where he is currently working toward the Ph.D. degree with the State Key Laboratory of Complex Electromagnetic Environment Effects.

His current research interests include object detection, image classification, few-shot learning and machine learning, and its applications to remote sensing images.

Caiguang Zhang received the B.S. degree in electronic information science and technology from the Qingdao University of Science and Technology, Qingdao, China, in 2017, and the M.S. degree in information and communication engineering in 2019 from the National University of Defense Technology, Changsha, China, where he is currently working toward the Ph.D. degree with the State Key Laboratory of Complex Electromagnetic Environment Effects.

His current research interests include interpretation in remote sensing images and computer vision, especially in object detection in optical remote sensing image.



Sinong Quan received the B.S. degree in mechanical and electronic engineering from the South China University of Technology, Guangzhou, China, in 2013, and the M.S. and Ph.D. degrees in information and communication engineering from the National University of Defense Technology, Changsha, China, in 2015 and 2019, respectively.

He is currently a Lecturer with the School of Electronic Science, National University of Defense Technology. His current research interests include polarimetric anti-interference and target detection, pattern recognition, synthetic aperture radar/polarimetric synthetic aperture radar image processing, and machine learning.

Dr. Quan was a recipient of the Outstanding Graduate Award from the National University of Defense Technology, in 2015.

Huaxin Xiao received the B.E. degree in automation from the University of Electronic Science and Technology of China, Chengdu, China, in 2012, and the Ph.D. degree in systems engineering from the National University of Defense Technology, Changsha, China, in 2018.

He is currently a Lecturer with the College of System Engineering, National University of Defense Technology. From 2016 to 2018, he was a Visiting Student with the National University of Singapore. His current research interests include saliency detection and image/video object segmentation.

Dr. Xiao received the Winner Prize of Object Localization Task at the Large Scale Visual Recognition Challenge 2017.

Gangyao Kuang (Senior Member, IEEE) received the B.S. and M.S. degrees in geophysics from the Central South University of Technology, Changsha, China, in 1988 and 1991, respectively, and the Ph.D. degree in communication and information from the National University of Defense Technology, Changsha, in 1995.

He is currently a Professor with the School of Electronic Science, National University of Defense Technology. His research interests include remote sensing, synthetic aperture radar (SAR) image processing, change detection, SAR ground moving target indication, and classification with polarimetric SAR images.

Li Liu (Senior Member, IEEE) received the B.Sc. degree in communication engineering, the M.Sc. degree in photogrammetry and remote sensing, and the Ph.D. degree in information and communication engineering from the National University of Defense Technology (NUDT), Changsha, China, in 2003, 2005, and 2012, respectively.

In 2012, she joined the faculty with the NUDT, where she is currently an Associate Professor with the College of System Engineering. During her Ph.D. research, she spent more than 2 years as a Visiting Student with the University of Waterloo, Waterloo, ON, Canada, from 2008 to 2010. From 2015 to 2016, she spent 10 months visiting the Multimedia Laboratory, Chinese University of Hong Kong, Hong Kong. From December 2016 to November 2018, she was a Senior Researcher with the Machine Vision Group, University of Oulu, Oulu, Finland. Her papers have currently more than 2500 citations in Google Scholar. Her current research interests include computer vision, pattern recognition, and machine learning.

Dr. Liu was a co-Chair for nine international workshops at the Conference on Computer Vision and Pattern Recognition, the International Conference on Computer Vision, and the European Conference on Computer Vision. She was a Guest Editor for special issues for the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE and the *International Journal of Computer Vision*. She serves as Area Chair for the 2020 Asian Conference on Computer Vision and the 2020 Integrated Computational Materials Engineering. She is an Associate Editor for the *Visual Computer Journal and Pattern Recognition Letter*.