Multi-Label Long-Tailed Datasets Classification: A survey and feature decoupling learning

The report for research project COMP 9993

School of Computer Science & Engineering

Faculty of Engineering

December 2021

by
Zhenxiang Lin (z5240946)
zhenxiang.lin@student.unsw.edu.au

Abstract

Naturally collected data generally shows a long-tailed distribution. In recent years, many methods have been proposed for long-tailed distribution problem, but most of them focus on single-label datasets. For multi-label problem, co-occurrence is a serious problem which has not been address well. In addition, different with single-label problem, multi-label loss functions based on BCE suffer from dominance of negative classes. The aim of this research is to summarize the proposed methods for multi-label classification problem in recent years and analyze their advantages. Meanwhile, the advantages are combined to generate a new loss function. Experiments show that the mixed methods with a good feature extractor can achieve a better performance. The best mAP for VOC-MLT and COCO-MLT are 91.14% and 69.63% respectively.

Keywords: Long-Tailed Distribution, Multi-Label Classification, co-occurrence, dominance of negative classes.

Table of Contents **Page**

1 Introduction

With the development of deep learning, the performance of many visual recognition tasks has a great improvement. However, most of them use balanced dataset. In real world, collected datasets cannot be absolutely balanced. In general, the data distribution is like a long tail. Some classes occupy a large proportion, while some classes are little. For example, in medical diagnosis field, most samples are negative samples. Positive samples only account for a little part. In addition, in anomaly detection field, anomaly samples are not much. In spite of this, these little proportion samples are still important. Moreover, although many researches have focused on long-tailed distribution problem, most of them use single-label datasets where each image only includes one instance [1,2,3,4]. However, in the open world, multiple instances could appear in a collected image. Meanwhile, a single instance may have multiple different features [5]. Thus, multi-label image recognition is also a foundational and important task in Computer Vision.

Currently, main type of loss function that used for multi-label classification task is binary cross entropy and its variants [6,7,8,9]. Sigmoid is its main activation function. There are two main challenges for multi-label classification task, dominance of negative labels and label co-occurrence.

For multi-label classification, many instances often appear together in an image. For example, when a bicycle appears in an image, it is greatly possible that human also appears. Nevertheless, dogs and birds seldom appear together. Different labels exist a relation. This relation leads to the label co-occurrence. Thus, after resampling, the distribution of classes still cannot be balanced.

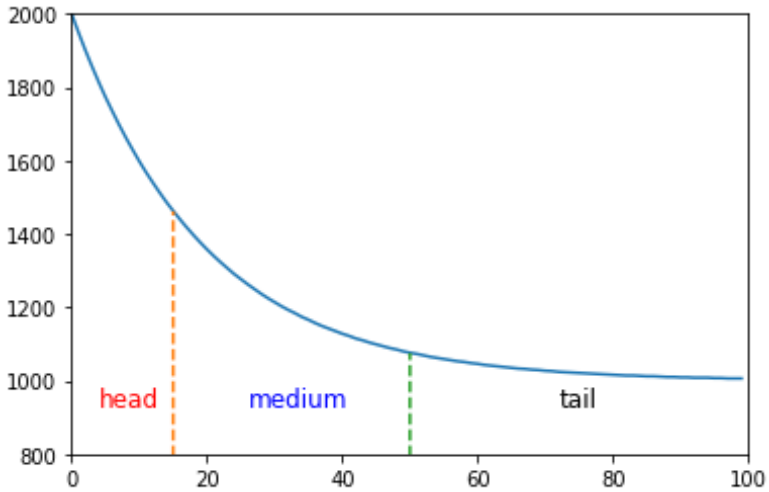

Fig. 1 Long-Tailed Distribution

In addition, for single-label classification, softmax is main activation function. It allows each label to exclude other labels. This is because there is only one label in an image. However, for multi-label problem, because different labels could appear in an image, excluding other

labels cannot be allowed. Thus, softmax is not appropriate for this problem. Binary cross entropy regards each class as a binary classification problem. For instance, if a dataset has 100 different classes, the problem will be split to 100 binary classification problem. Nonetheless, this leads to a situation. In a multi-label image, most labels are negative, but binary cross entropy symmetrically considers positive and negative labels. Negative labels will contribute much more than positive labels, so the loss will be dominated by negative labels. Therefore, it is also a vital problem which needs to be solved.

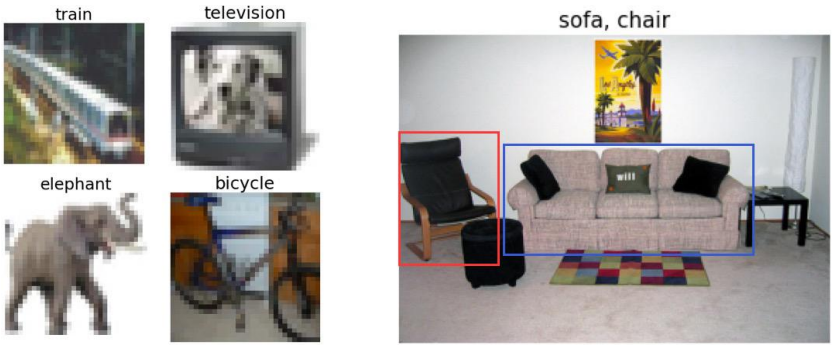

Fig. 2 Difference with single-label images (left) and multi-label images (right)

Furthermore, feature extraction is always an important task. CNN and its variants [20,21] have a great effect on computer vision. With the proposition of Vision Transformer [22], the performances of different tasks are improved. Nonetheless, it needs a great size of dataset and long time to train, which prevent its development. Recently, some approaches [23,24] are proposed to overcome its flaws and meanwhile, keep its high performance.

In this report, some useful methods are summarized. These methods are based on pre-process, multi-label co-occurrence, dominance of negative classes and feature extraction. By analyzing their advantages and disadvantages, the methods will be combined to obtain a better performance.

Furthermore, according to the conclusion, a feature decoupling learning strategy is proposed. After analyzing the results, I summarize some shortcomings and suggest some possible future works.

2 Literature review

2.1 Long-tailed Problem

Currently, many researches for long-tailed distribution problem can be spilt by sampling strategy, reweighting strategy and model structure [1,4,10]. Some of them can also be applied in multi-label classification problem.

2.1.1 Sampling Strategy

There are 3 sampling strategy, instance sampling, class balanced sampling and progressively balanced sampling [11].

Instance sampling is a very common sampling strategy. Each image in the dataset will be sampled with an equal probability. The probability P_I is

$$P_I = \frac{1}{N}, \tag{1}$$

where N is the size of dataset.

The aim of class balanced sampling is to make data distribution balanced. It can be split by over-sampling and under-sampling. Over-sampling repetitively samples data from minority classes. Under-sampling discards a part of majority classes. Although they can rebalance the distribution, there are still some disadvantages. Over-sampling may cause overfitting of minority classes, and under-sampling will lose information about majority classes. Moreover, a class-aware sampling is used in [12]. By this approach, the probability of each class which appears in a batch is equal. Meanwhile, the order of input images is different. Ideally, after class balanced sampling, the sampling probability of each class P_C should be

$$P_C = \frac{1}{C} \tag{2}$$

and the sampling probability of each image P_k should be

$$P_k = \frac{1}{C} * \frac{1}{N_k}, \tag{3}$$

where C is the number of classes and N_k is the number of the k-th class. However, due to the co-occurrence of multi-label problem, the distribution cannot be absolutely balanced.

Progressively balanced sampling combines both sampling strategies. It is a mixed sampling strategy. It firstly uses instance sampling, but changes to class balanced sampling gradually during training. Its sampling probability of each image P_k can be calculated as

$$P_k = \left(1 - \frac{t}{T}\right) * \frac{1}{N} + \frac{t}{T} * \frac{1}{C} * \frac{1}{N_k}, \tag{4}$$

where T is the number of total training epochs and t is current training epoch.

2.1.2 Reweighting

Reweighting is a loss rebalance strategy based on loss function. It mainly decreases the weight of majority classes and increases the weight of minority classes. In general, the weight is decided by the distribution of dataset. Its value is based on the inverse of each classes size [1,13]. The purpose of reweighting is similar with class balanced sampling which is to rebalance the contribution of each class. However, co-occurrence problem will still have influence on its performance.

2.1.3 Model Structure

In recent years, some new models are proposed to deal with long-tail distribution problem. Dual-branches model and its variants can consider two different distributions at the same time [4,14,15]. Each branch can learn different feature by different input distribution. Two generated features are combined by the control. This method can manually adjust the input data distribution and parameters of controller to balance the effect of each class, but it is still difficult to decrease the effect caused by co-occurrence problem.

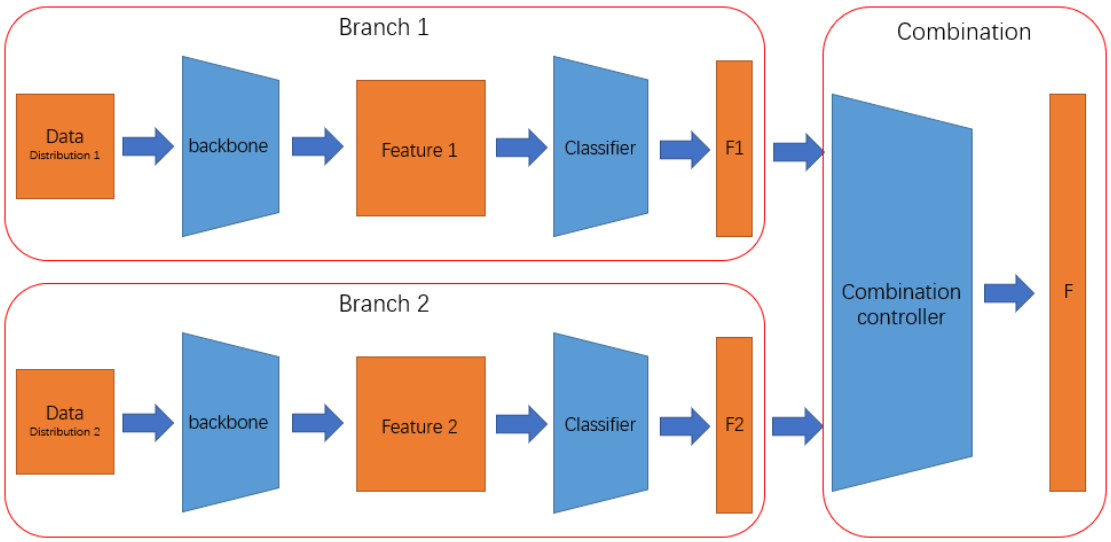

Fig. 3 Structure of dual-branches model

2.2 Multi-Label Problem

There are two challenges for multi-label problem, co-occurrence and dominance of negative labels.

2.2.1 co-occurrence

Some methods are proposed to address multi-label co-occurrence problem. One is based on loss function [16]. It aims to calculate the practical appearance probability of each class. The other method is to use graph convolution network [8,17]. It uses GCN to address the correlation of labels and CNN to extract feature of images. However, because of long-tailed distribution of dataset, the distribution of training set and test set is quite different, which means their co-occurrence rates are still different. This could decrease the performance of the model.

2.2.2 dominance of negative labels

Currently, in order to deal with this problem, main approaches focus on decrease the weight of negative labels. Regularization [16] and threshold control [18] can directly reduce the contribution of negative labels. In addition, focal loss [19] and its variants [9] also have a positive effect on this problem.

2.3 Feature extraction

General methods still use ResNet [21] and its variants as backbone to extract feature of images. However, in recent years, some research about the combination of traditional convolution and Transformer have shown a better performance on computer vision tasks, such as Swin-Transformer [24] and Nested Transformer [23].

3 Method

In this section, I will introduce different methods used for multi-label or long-tailed distribution problem. I will also explain their principle. At last, I will introduce a feature decoupling learning strategy that I design.

3.1 Reweighting Method

3.1.1 Conventional Reweighting Loss Function

A common reweighting strategy [13,15] is to use the ratio of positive samples of each class to calculate the weight of each class. Then, this weight ω_k can be incorporated into binary cross entropy loss function. The loss function becomes

$$Loss = -\frac{1}{C}\sum_{k=1}^{C} \omega_k^i * \left(y_k^i * \log\left(\sigma(x_k^i)\right) + \left(1 - y_k^i\right) * \log\left(1 - \sigma(x_k^i)\right)\right) \tag{5}$$

and the weight ω_k^i is

$$\omega_k^i = y_k^i * e^{1-\rho_k} + \left(1 - y_k^i\right) * e^{\rho_k}, \tag{6}$$

where C is the number of classes in the dataset, x_k^i is the k-th logit and y_k^i is the k-th ground-truth label of the i-th sample. σ is sigmoid function and ρ_k is the ratio of positive samples for the k-th class.

$$\rho_k = \frac{N_k}{N} \tag{7}$$

3.1.2 Distribution-Balanced Reweighting Loss Function

For multi-label long-tailed classification problem, the distribution after class-balanced sampling still cannot be balanced. Some head classes still occupy a large proportion. This is because of the co-occurrence of multi-label problem. If the co-occurrence problem is not taken into consideration, the sampling probability of each sample after class balanced sampling is shown in Eq. (3). However, in fact, the appearance of i-th class may be not because it is sampled but the j-th class. This is due to the conditional probability $p(i|j)$ which is shown as Eq. (8).

$$p(i|j) = \frac{N_{i\cap j}}{N_j} \tag{8}$$

Distribution-Balanced Reweighting [16] combines class-level sampling frequency and instance sampling frequency. Class-level sampling frequency $P^C(x_k^i)$ does not consider co-occurrence, but instance-level sampling frequency $P^I(x_k^i)$ takes it into consideration. Their formulas are shown in Eq. (9). The weight is the ratio of two frequency, which is shown in Eq. (10).

$$P^C(x_k^i) = \frac{1}{C} * \frac{1}{N_k}, \qquad P^I(x_k^i) = \frac{1}{C} * \sum_{y_k^i=1} \frac{1}{N_k} \tag{9}$$

$$r_k^i = \frac{P^C(x_k^i)}{P^I(x_k^i)} \tag{10}$$

In order to avoid the occurrence of zero weight, a mapping function is designed. Its formula is

$$\omega_k^i = \alpha + \frac{1}{1 + e^{-\beta * (r_k^i + \mu)}}, \tag{11}$$

where α is used to adjust the range of its value. β and μ can adjust the convergence speed of mapping function.

The last format of loss function is same as Eq. (5).

3.2 Negative Labels Suppression

3.2.1 Adaptive Class Suppression

A common approach to reduce the effect is to ignore them directly [18]. It firstly sets a threshold ξ whose range is 0 to 1. If the probability of a negative label p_k^i is greater than this threshold, this means the model still cannot classify this label clearly. The weight is set as 1. Otherwise, the weight is set as 0 to avoid the dominance of negative labels. The formula of ω_k^i is shown by Eq. (12) and the probability p_k^i is calculated by sigmoid function.

$$\omega_k^i = \begin{cases} 1, & \text{if } k \text{ is positive class} \\ 1, & \text{if } k \text{ is negative class and } p_k^i \geq \xi \\ 0, & \text{if } k \text{ is negative class and } p_k^i < \xi \end{cases} \tag{12}$$

Moreover, this approach can be combined with reweighting strategy. After combination, the weight becomes

$$\omega_k^i = \begin{cases} \omega_k^i, & \text{if } k \text{ is positive class} \\ \omega_k^i, & \text{if } k \text{ is negative class and } p_k^i \geq \xi \\ 0, & \text{if } k \text{ is negative class and } p_k^i < \xi \end{cases}. \tag{13}$$

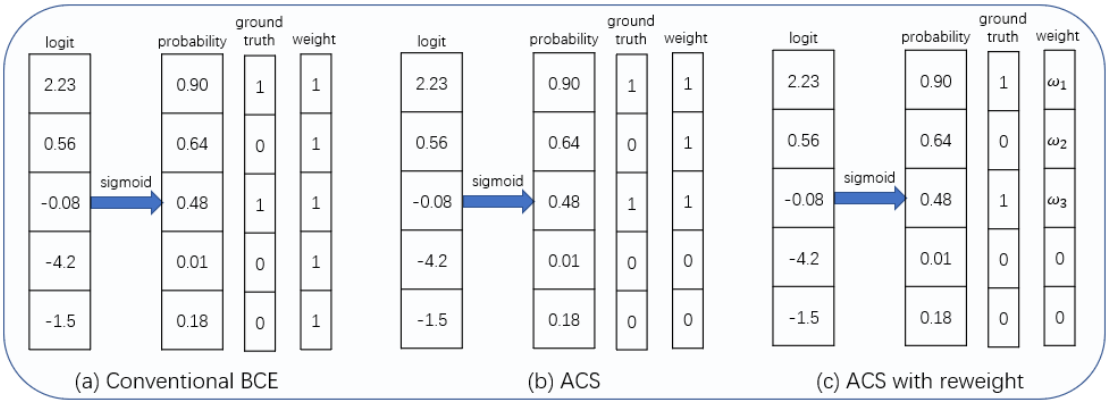

Fig. 4 Weight after using different methods. (a) conventional binary cross entropy, (b)

original adaptive class suppression, (c) adaptive class suppression with reweight

3.2.2 Negative-Tolerant Regularization

Regularization is also a feasible method to avoid dominance of negative labels [16]. It uses a parameter λ to linearly reduce the contribution of negative labels and a parameter v as class-specific bias. Its unique format is

$$Loss_{NT} = -\frac{1}{C}\sum_{k=1}^{C} y_k^i \log\left(\sigma\left(x_k^i - v_k\right)\right) + \frac{1}{\lambda}\left(1 - y_k^i\right)\log\left(1 - \sigma\left(\lambda\left(x_k^i - v_k\right)\right)\right), \tag{14}$$

$$v_k = -\log\left(\frac{N}{N_k} - 1\right) * \frac{v}{\lambda}. \tag{15}$$

After the combination with reweight, its format becomes

$$Loss = -\frac{1}{C}\sum_{k=1}^{C} \omega_k^i \left(y_k^i \log\left(\sigma\left(x_k^i - v_k\right)\right) + \frac{1}{\lambda}\left(1 - y_k^i\right)\log\left(1 - \sigma\left(\lambda\left(x_k^i - v_k\right)\right)\right)\right). \tag{16}$$

3.3 Focal

3.3.1 Conventional Focal Loss

Focal loss [19] is designed to solve imbalanced distribution problem. Its aim is to reduce the weight of a large of negative labels. This is appropriate for multi-label long-tailed distribution problem. Its format is

$$Loss = -\frac{1}{C}\sum_{k=1}^{C} y_k^i \alpha\left(1 - p_k^i\right)^{\gamma} \log\left(\sigma\left(x_k^i\right)\right) + \left(1 - y_k^i\right)\alpha\left(p_k^i\right)^{\gamma} \log\left(1 - \sigma\left(x_k^i\right)\right), \tag{17}$$

where γ is the factor to adjust the focal effect and α is a balanced parameter.

In addition to these advantages, focal loss takes more attention on the samples that are difficult to be classified.

3.3.2 Asymmetric Focal Loss

Focal loss uses a symmetric γ for both positive and negative classes. Asymmetric Focal Loss [9] divides them into two different parameters which are γ^+ and γ^-. In general, γ^- is larger than γ^+ to decrease more effects of negative labels. If γ^+ is set as 0, then the effects of positive classes will be all reserved.

$$Loss = -\frac{1}{C}\sum_{k=1}^{C} y_k^i * \left(1 - p_k^i\right)^{\gamma^+} \log\left(\sigma\left(x_k^i\right)\right) + \left(1 - y_k^i\right) * \left(p_k^i\right)^{\gamma^-} \log\left(1 - \sigma\left(x_k^i\right)\right) \tag{18}$$

Focal, negative labels suppression methods and reweighting methods can be combined to deal with multi-label long-tailed classification problem.

3.4 Swin-Transformer

Swin-Transformer [24] adopts the idea of both Vision Transformer and CNN. It is a hierarchical structure model. It uses patch partition and patch merging to down-sample, which is similar with the convolution whose kernel size and stride are equal. For patch partition, its window size is 4, which means an image whose dimension is $(224, 224)$ can be segmented to 16 $(56, 56)$ sub-images. For patch merging, the window size is 2.

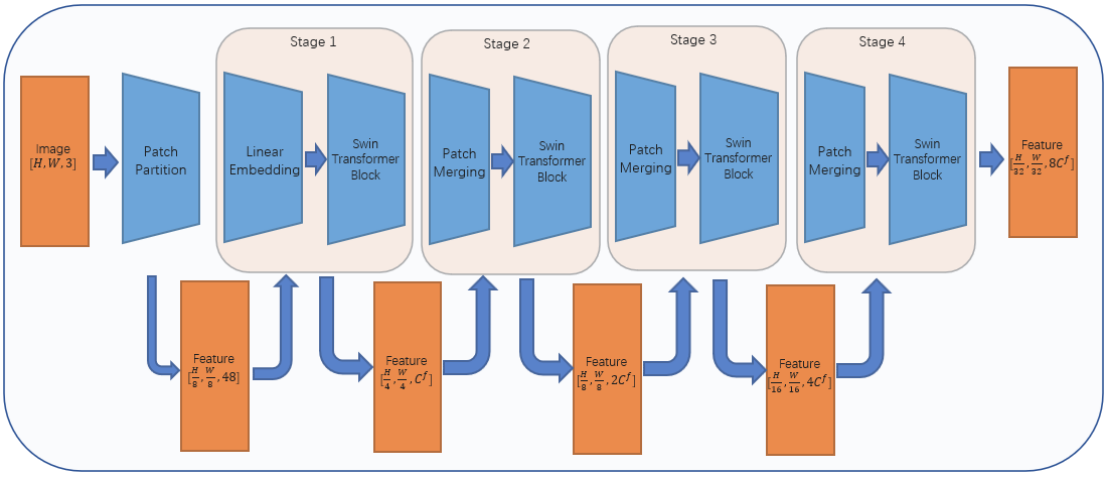

Fig. 5 Architecture of Swin-Transformer

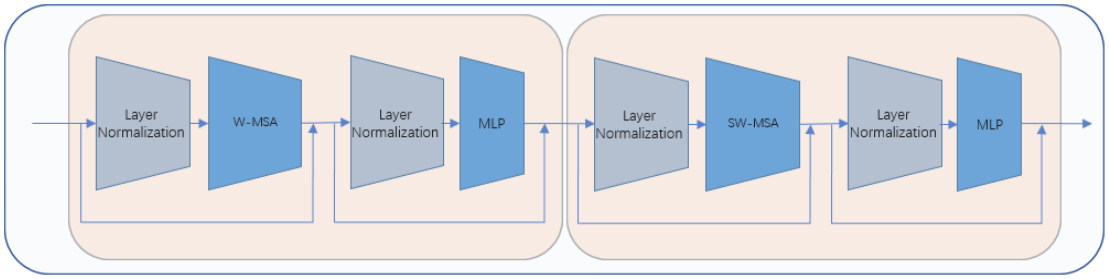

Fig. 6 Architecture of Swin-Transformer Block.

Its architecture is like ResNet, which uses 4 stages. ach stage uses a patch partition and some swin-transformer blocks. A swin-transformer block is constructed by a window based multi-head self-attention (W-MSA) and a shifted window based multi-head self-attention (SW-MSA) as well as some other modules.

W-MSA calculates the attention in a single window. This has a problem that each window can only compute its own information. However, shifted window can alleviate this problem. The windows are moved by a few pixels, which allows information interaction among windows. Therefore, the use of both W-MSA and SW-MSA can expand the range of information perception.

3.5 Feature Decoupling Learning Strategy

3.5.1 Model Design

According to the conclusion of above methods, I design a feature decoupling learning

strategy.

Traditionally, the process of multi-label classification is to use a backbone to extract image feature and a classifier to generate k logits. Then, use sigmoid function to calculate the existence probability of k labels. This is a feasible method that has been confirmed, while co-occurrence problem has an influence on the classification performance. For example, if due to data collection problem, cars and drivers always appear together in an image, then a traditional training model will take this co-occurrence into consideration. However, in fact, drivers are humans. In terms of image features, there is no similarity between humans and cars. Therefore, an ideal classifier should be able to divide the features of each class from the entire image feature.

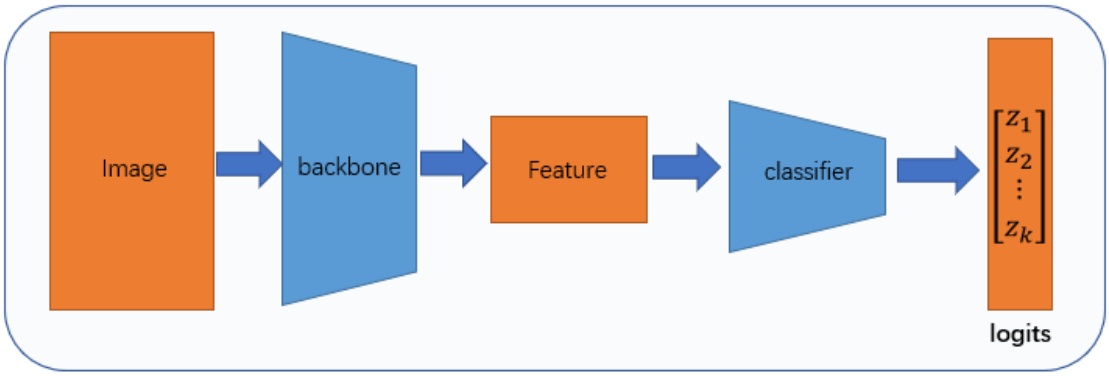

Fig. 7 Traditional training strategy

Based on this theory, I design a new training strategy which is shown in figure 8. After the generation of image feature, it uses a decoupler module rather than a classifier. The generated feature includes all information in an image. However, some information may be not useful for a specific classification task. For example, if a task needs to know whether an image includes cats or dogs, tables and other background instances are useless. Only the features of dogs and cats need to be extracted. All other information can be given up. Therefore, the decoupler module can decouple the image feature and generate k vectors. Each vector c_k represents the feature of a specific object. Each class representation vector v_k is a standard vector of an objects. If c_k is similar with v_k, then it is greatly possible that the object of class k is in the image. If c_k is greatly different with v_k, then the object of class k is not in the image

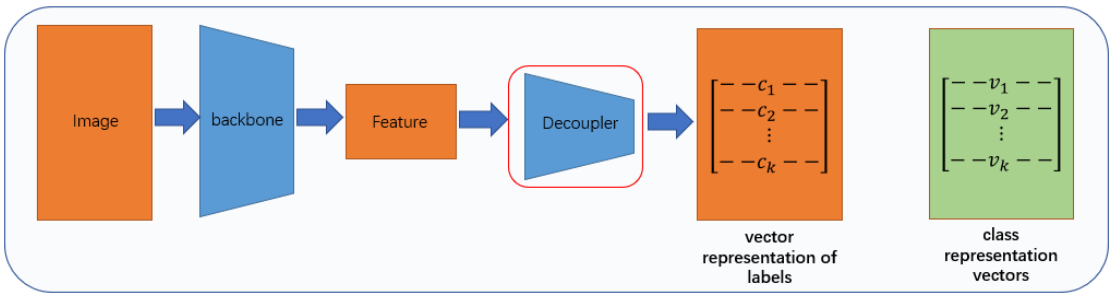

Fig. 8 Feature decoupling learning strategy

3.5.2 Decoupler

There are two different decoupler that I design.

3.5.2.1 Multi-Layer Perceptron Decoupler

The shape of features is (C^f, H, W). Thus, firstly, the feature needs to be flattened. Then, through a multi-layer perceptron module, the feature can be decoupled. At last, the shape of generated feature can be reshaped to (C, D). D is the dimension of class representation vector and C is the number of classes. C^f is the number of channels of the feature. H and W are height and weight of the feature.

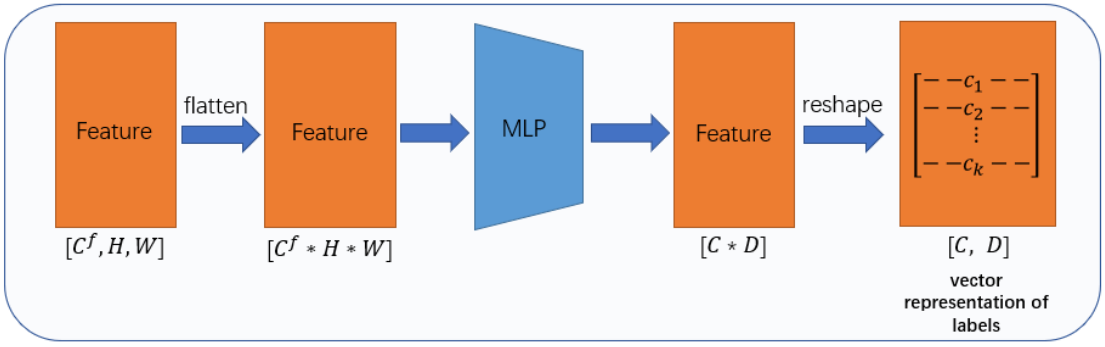

Fig. 9 Structure of Multi-Layer Perceptron Decoupler

3.5.2.2 Matrix Decoupler

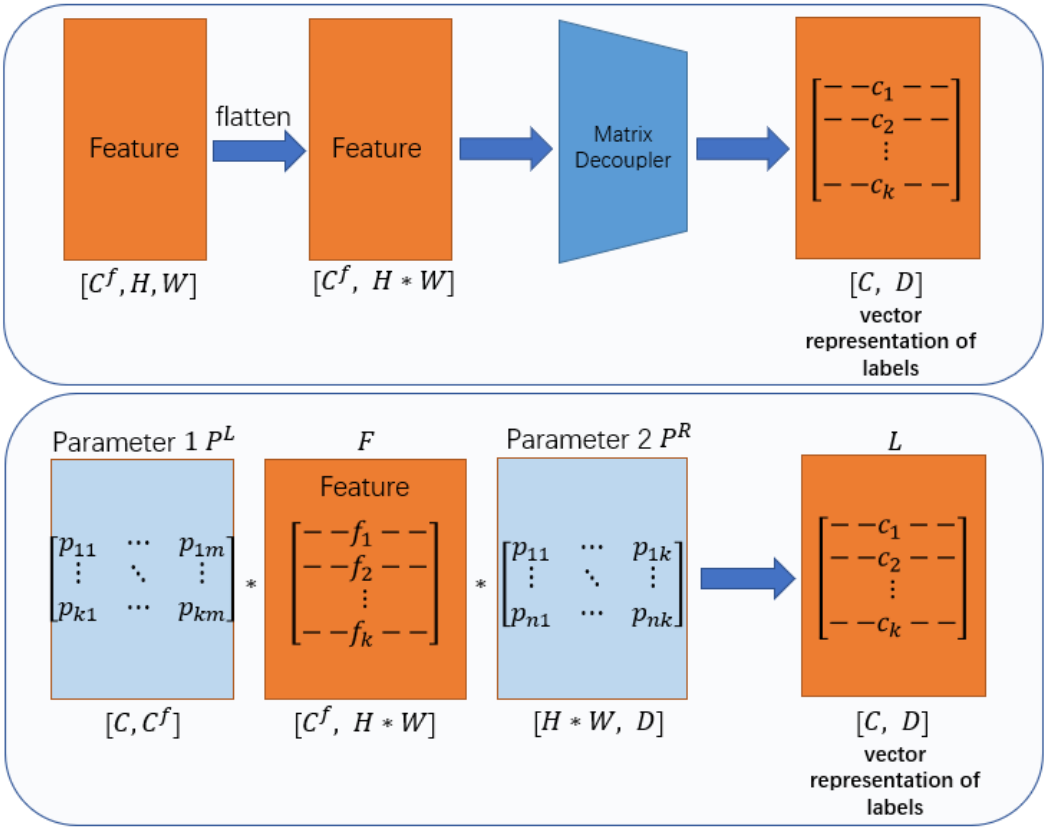

Fig. 10 Structure of Matrix Decoupler Module

The other decoupler is matrix decoupler. Different with multi-layer perceptron decoupler, its shape of feature is flattened to $(C^f, H*W)$. Then, through a matrix decoupler to generate label representation vectors directly. The decoupler uses two learnable matrices as parameters. By the product of three matrices, the feature can be decoupled. The label representation vectors matrix L can be calculated by

$$L = P^L * F * P^R, \tag{19}$$

where P^L is parameter 1, P^R is parameter 2 and F is feature matrix.

3.5.3 Vector Representation Approach

3.5.3.1 One-hot Encoding Representation

One-hot encoding is one of the most common representation approaches. Each representation vector has only one valid value which is one and the rest are zero. For this approach, loss function still needs adjustment.

The idea of feature decoupling learning strategy is to make label vectors whose instance appears in the image and the class representation vectors as similar as possible. Those labels that do not appear are to be as different as possible. Therefore, using vector inner product to calculate their similarity could be appropriate. Initially, the similarity z_k is

$$z_k = \sum_{i=0}^{D} c_{ki} * v_{ki}\,. \tag{20}$$

Because of the feature of one-hot encoding, its format become

$$z_k = c_{kk}. \tag{21}$$

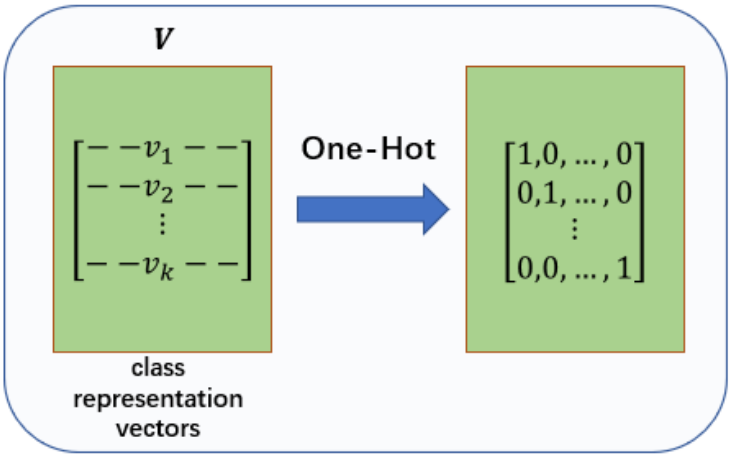

Fig. 11 One-hot encoding representation of class representation vectors

According to Eq. (20), if c_k and v_k are similar, z_k should be large. If they are greatly different, z_k should be greatly less than zero. However, the range of c_k is too large. Thus, a softmax function is used on L to make the range of c_k from 0 to 1. It can be shown as Eq. (22).

$$z_k = \mathrm{softmax}(c_k)_k \tag{22}$$

Then, the loss function is

$$Loss = -\frac{1}{C}\sum_{k=1}^{C} y_k^i * \log(z_k) + \left(1 - y_k^i\right) * \log(1 - z_k) \tag{23}$$

3.5.3.2 Word-vector Representation

Inspired by MLGCN [8], word-vector can be used to address the relation among labels. Therefore, I try to use word-vector as class representation vectors.

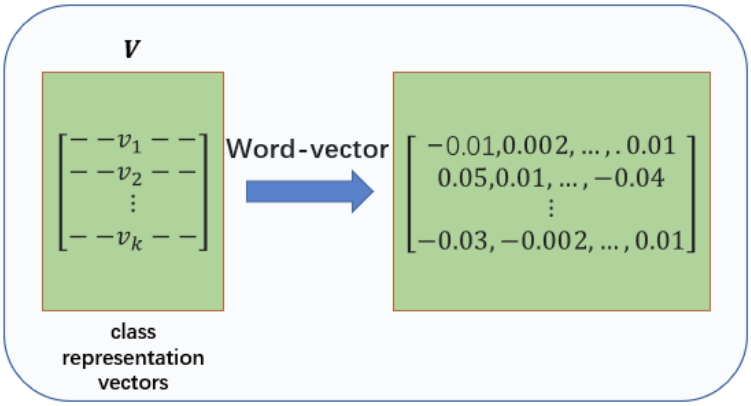

Fig. 12 Word-vector representation of class representation vectors

Because the value of word-vector is not binary and its value is not all positive, softmax function is not appropriate. In addition, because the range of vector inner product is -∞ to +∞, it cannot be used into loss function directly. Thus, I use cosine similarity to measure them. The similarity z_k is

$$z_k = \frac{\sum_{i=0}^{D} c_{ki} * v_{ki}}{\sqrt{\sum_{i=0}^{D} {c_{ki}}^2} * \sqrt{\sum_{i=0}^{D} {v_{ki}}^2}}. \tag{24}$$

Its range is -1 to 1. If two vector is similar, then z_k should be close to 1, otherwise it should be less than or equal to 0.

However, there still exists a problem that z_k cannot be used into binary cross entropy directly. Consequently, two approaches can adjust the range to 0 to 1, which are shown in Eq. (25) and Eq. (26).

$$z_k = \begin{cases} 0, & \text{if } z_i < 0 \\ z_k, & \text{otherwise} \end{cases} \tag{25}$$

$$z_k = \frac{z_k + 1}{2} \tag{26}$$

Then, the loss function can be

$$Loss_{cos} = -y_k^i * \log(z_k) - \left(1 - y_k^i\right) * \log(1 - z_k) \tag{27}$$

The cosine similarity can measure the angle between two vectors, but it cannot measure their norms. Thus, their similarity of norms should also be considered. Its loss is shown in Eq. (28).

$$Loss_{\text{norm}} = y_k^i * (c_k - v_k)^2 \tag{28}$$

Then, use a control parameter γ to adjust the weight of $Loss_{\text{norm}}$. Therefore, the final format of loss function is

$$Loss = -\frac{1}{C}\sum_{k=1}^{C} y_k^i * \log(z_k) + \left(1 - y_k^i\right) * \log(1 - z_k) + \gamma * y_k^i * (c_k - v_k)^2 \tag{29}$$

4 Experiments

4.1 Dataset

Followed by [16], I conduct the experiments on VOC-MLT and COCO-MLT. They are respectively extracted from the VOC [25] and COCO [26] datasets.

4.1.1 VOC-MLT dataset

VOC-MLT training set includes 1,142 images from VOC2012. Its test set includes 4,952 images from VOC2007. There are totally 20 classes. The most class which is human included 775 images, while the least class which is cow has only 4 images. Classes whose number of images are more than 100 are head classes. Those whose number of images are less than 20 are tail classes. The rests are medium classes. The ratio of head, medium and tail classes is 6:6:8. Among all classes, human has a high probability of appearing with other classes.

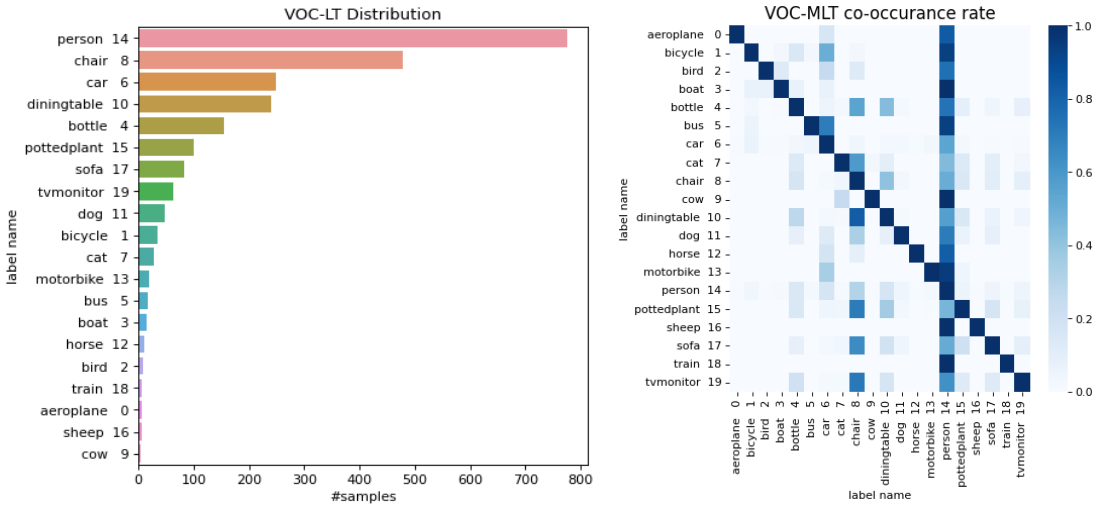

Fig. 13 Distribution of VOC-MLT (left) and co-occurrence rate of VOC-MLT (right)

4.1.2 COCO-MLT dataset

COCO-MLT training set includes 1,909 images from COCO2017. Its test set includes 5,000 images from COCO2017. There are totally 80 classes. The most class which is person included 1,128 images, while the least class which is hair drier has only 6 images. The ratio of head, medium and tail classes is 22:33:25. Among all classes, person has the highest probability of appearing with other classes.

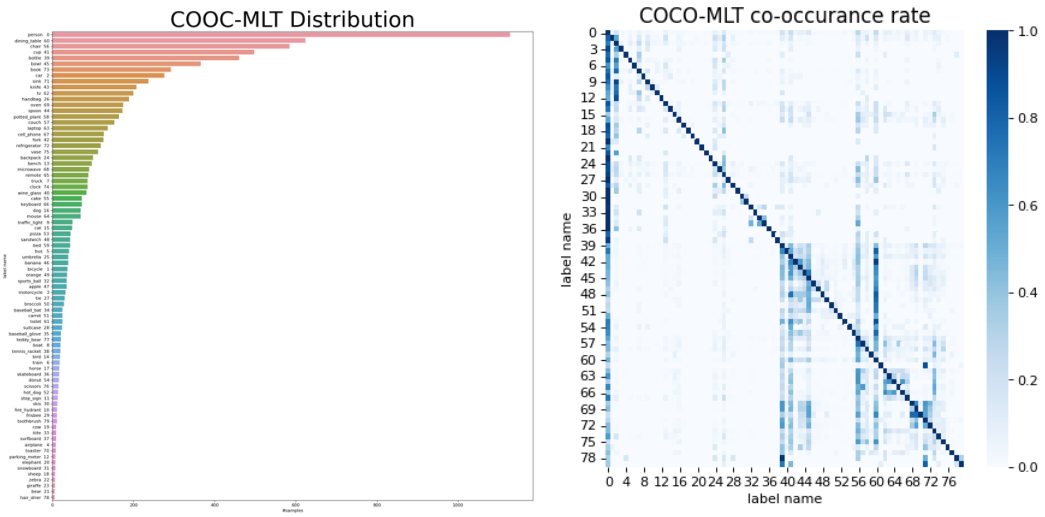

Fig. 14 Distribution of COCO-MLT (left) and co-occurrence rate of COCO-MLT (right)

In addition, due to the different size of each image for VOC and COCO, the width and height of all images are resized to 224.

4.2 Evaluation Metrics

The distribution of test data is not absolutely balanced. Their distributions are similar with the original VOC and COCO. Therefore, mean average precision (mAP) is used to evaluate the performance of different methods.

$$mAP(Classes) = \frac{1}{\#\ Classes}\sum AP(each\ class) \tag{30}$$

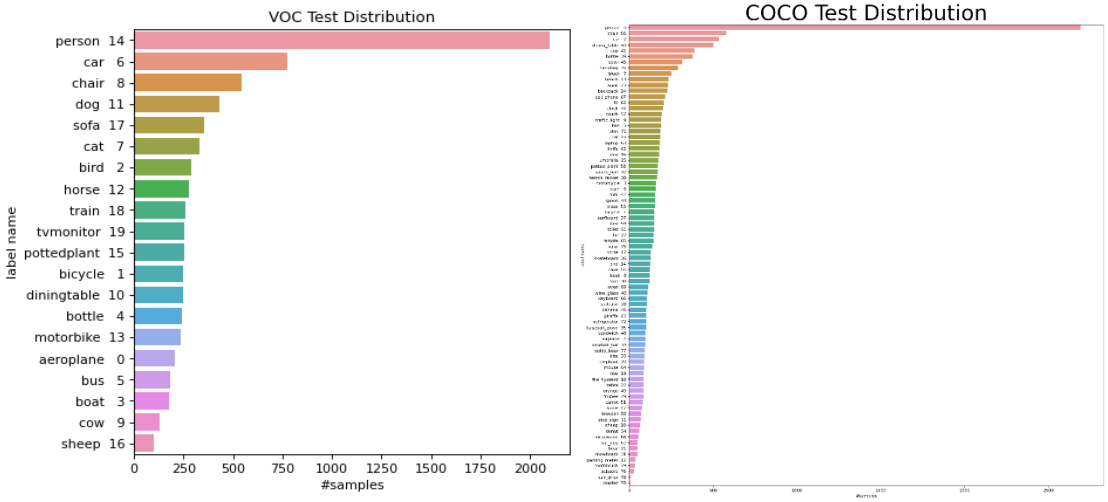

Fig. 14 Distribution of COCO-MLT (left) and co-occurrence rate of COCO-MLT (right)

4.3 Implementation Configuration

In my experiment, all images are resized to $(224,224)$ firstly. The optimizer is stochastic gradient descent with 0.9 momentum and 10^{-4} weight decay. Initial learning rate is 0.02. After some steps, the learning rate is divided by 10. A linear warm-up strategy is also used

for first 500 iterations. Its factor is $\frac{1}{3}$.

For ResNet, there are totally 8 training epochs on VOC-MLT and 40 epochs on COCO-MLT. For VOC-MLT, after 5 and 7 epochs, the learning rate is decreased and for COCO-MLT, after 25 and 35 epochs, the learning rate is decreased.

For Swin-Transformer, there are totally 14 epochs on VOC-MLT and 50 epochs on COCO-MLT. For VOC-MLT, after 8 and 12 epochs, the learning rate is decreased and for COCO-MLT, after 30 and 45 epochs, the learning rate is decreased. Each epoch includes 121 iterations and batch size is 32.

4.4 Ablation Study

4.4.1 Augmentation Methods

Firstly, I use different augmentation strategies to find the best combination. They include horizontal flip, brightness transformation, hue transformation, saturation transformation, contrast transformation, affine transformation, random channel swap and random crop.

In this step, loss function is binary cross entropy and backbone is ResNet50 pre-trained on ImageNet. Sampling strategy is class aware sampling. Dataset is VOC-MLT.

The best combination is using horizontal flip, brightness transformation, hue transformation, saturation transformation, contrast transformation, and random crop. Its mAP 77.06. This combination is applied in subsequent experiments.

Table 1: Results of different augmentation strategies on VOC-MLT

Flip	brightness	hue	saturation	contrast	affine	channel swap	crop	mAP
√								75.7
√	√							75.88
√	√	√						76.11
√	√	√	√					76.14
√	√	√	√	√				76.37
√	√	√	√	√	√			74.85
√	√	√	√	√		√		74.88
√	√	√	√	√			√	77.06

4.4.2 Reweighting Methods

After the decision of the augmentation combination, two reweighting strategies are compared.

Table 2 shows the results of two different reweighting strategy. For VOC-MLT, mAP of distribution-balanced reweighting is less than conventional reweighting strategy, while for COCO-MLT, distribution-balanced reweighting has a higher performance. In summary, there is no great difference between these two methods.

Table 2: Comparison results of conventional reweighting and distribution-balanced reweighting

VOC-MLT				
Reweight	mAP	mAP (head)	mAP (medium)	mAP (tail)
BCE	77.49	77.81	83.20	71.37
R-BCE	77.74	78.46	82.96	71.57
DB-BCE	77.69	79.78	81.93	70.66
COCO-MLT				
Reweight	mAP	mAP (head)	mAP (medium)	mAP (tail)
BCE	47.74	49.19	50.26	43.15
R-BCE	50.27	48.33	53.30	47.97
DB-BCE	50.62	47.81	53.29	49.57

* R-BCE is conventional reweighting and DB-BCE is distribution-balanced reweighting

4.4.3 Negative Labels Suppression Methods

In this part, distribution-balanced reweighing is used to be combined with two methods.

For adaptive class suppression, threshold ξ is set as 0.2 for both VOC-MLT and COCO-MLT. For negative-tolerant regularization, negative parameter λ is set as 5 and initial bias ν is set as 0.05 for VOC-MLT. However, for COCO-MLT, λ is set as 2 and initial bias ν is set as 0.05.

After the use of adaptive class suppression, mAP of VOC-MLT and COCO-MLT can be increased to 78.61% and 50.42%, respectively. After the use of negative-tolerant regularization, mAP can be increased to 79.75% and 51.40%, respectively.

Table 3: Comparison results of adaptive class suppression and negative-tolerant regularization

VOC-MLT				
suppression	mAP	mAP (head)	mAP (medium)	mAP (tail)
ACS	78.61	72.46	83.66	79.44
NTR	79.75	73.72	83.87	81.18
COCO-MLT				
suppression	mAP	mAP (head)	mAP (medium)	mAP (tail)
ACS	50.42	49.87	52.75	47.54
NTR	51.40	50.18	54.18	48.62

* ACS is adaptive class suppression and NTR is negative-tolerant regularization

4.4.4 Focal Methods

For traditional focal loss, focal parameter γ is set as 2 and balanced parameter α is set as 2. For asymmetric focal, positive focal parameter γ^+ is set as 0 in order to emphasis the effects of positive labels and negative focal parameter γ^- is set as 3 to reduce their effects.

According to table 4, it can be seen that focal loss has a negative effect on the performance, while with asymmetric focal and NTR, mAP can achieve 79.78% and 51.20%,

respectively.

Table 4: Comparison results of focal and asymmetric focal

VOC-MLT					
suppression	focal	mAP	mAP (head)	mAP (medium)	mAP (tail)
ACS	Focal	77.29	70.48	80.35	80.10
	A- Focal	79.60	70.89	83.02	79.84
NTR	Focal	77.84	70.90	80.21	81.27
	A- Focal	79.78	74.02	83.68	81.17
COCO-MLT					
suppression	focal	mAP	mAP (head)	mAP (medium)	mAP (tail)
ACS	Focal	50.79	52.20	54.57	43.52
	A-Focal	50.38	49.61	52.39	48.24
NTR	Focal	45.43	52.06	46.39	36.46
	A- Focal	51.20	50.98	53.22	48.40

* Focal is traditional focal loss and A-Focal is asymmetric focal.

4.4.5 Swin-Transformer

Table 5: Result of Swin-Transformer on VOC-MLT

Backbone	Loss function	mAP	mAP (head)	mAP (medium)	mAP (tail)
ResNet 50	DB-NTA	79.78	74.02	83.68	81.17
Swin-T (Tiny)	BCE	80.49	67.23	82.49	88.93
Swin- T (Small)	BCE	85.99	77.51	85.97	92.37
Swin-T (Base)	BCE	89.97	85.36	88.79	94.94
Swin-T (Large)	BCE	90.21	85.09	89.67	95.16

* DB-NTA is distribution-balanced reweighting with negative-tolerant regularization and asymmetric focal loss. BCE is original binary cross entropy.

In this part, Swin-Transformer is used to compare with ResNet. There are 4 different structures, tiny, small, base and large. Their network parameters progressively increase in size. The result of ResNet 50 with DB-NTA is used as comparison. After the use of Swin-Transformer, the performance can achieve 90.21% on VOC-MLT.

Table 6 shows the result of Swin-Transformer with other methods. I use Swin-Transformer (Base) as backbone and the mAP can achieve 91.14% on VOC-MLT and 69.63% on COCO-MLT.

Table 6: Result of Swin-Transformer (Base) with other methods

VOC-MLT						
reweight	suppression	focal	mAP	mAP (head)	mAP (medium)	mAP (tail)
baseline (BCE)			89.91	81.48	92.28	94.45
R-BCE	NTR	A-Focal	90.77	82.02	92.53	96.01
		Focal	89.79	80.79	91.69	95.10
	ACS	A-Focal	90.67	82.09	92.29	95.89
		Focal	89.59	80.46	91.47	95.05
DB-BCE	NTR	A-Focal	90.62	81.20	92.61	96.19
		Focal	88.62	76.66	91.16	95.68
	ACS	A-Focal	91.14	82.54	92.75	96.39
		Focal	85.94	73.79	86.98	94.26
COCO-MLT						
reweight	suppression	focal	mAP	mAP (head)	mAP (medium)	mAP (tail)
baseline (BCE)			64.51	60.94	68.66	62.15
R-BCE	NTR	A-Focal	69.63	64.74	73.84	68.37
		Focal	68.94	63.87	72.90	68.17
	ACS	A-Focal	69.33	64.71	73.32	68.15
		Focal	68.21	63.19	71.98	67.66
DB-BCE	NTR	A-Focal	69.16	61.32	73.78	69.95
		Focal	65.73	53.30	70.57	70.28
	ACS	A-Focal	68.33	60.47	73.16	68.87
		Focal	67.78	57.09	73.59	69.52

* Baseline is original binary cross entropy.

4.4.6 Feature Decoupling Learning

Firstly, one-hot encoding representation is used as class representation vectors. Backbone is Swin-Transformer (Base). Dataset is VOC-MLT. Its mAP is 62.78% by matrix decoupler.

Table 7: Result of feature decoupling learning using one-hot encoding representation on VOC-MLT

Decoupler	mAP	mAP (head)	mAP (medium)	mAP (tail)
MLP	55.85	42.61	63.75	59.85
Matrix	62.78	42.78	69.56	72.70

Then, word-vector representation is conducted. control parameter γ is set as 0, which means we only consider cosine of vectors firstly. However, after 9 epochs, the loss does not change too much, which is around 0.6724. Its mAP keeps 7.71% which never improves during training.

5 Discussion

5.1 Augmentation

Augmentation is an important approach to compensate the lack of data. Nevertheless, not all augmentation strategies can improve the performance of deep learning model. It can be seen from table 1 that after using affine transformation and random channel swap, there is a reduction of mAP. Therefore, the appropriate combination of augmentation methods is significant.

5.2 Reweighting

According to table 2, reweighting strategy can improve the performance. For VOC-MLT, the improvement is not too much, but for COCO-MLT, there is a nearly 3% increase. However, the difference between conventional reweighting and distribution-balanced reweighting is not too much. For VOC-MLT, conventional reweighting leads by 0.05%, but for COCO-MLT, distribution-balanced reweighting leads by 0.35%. In summary, the improvement of using distribution-balanced reweighting is not great.

5.3 Negative Labels Suppression

Compared with adaptive class suppression, the performance of negative-tolerant regularization is better. Its mAP leads by nearly 1% on both VOC-MLT and COCO-MLT. Both methods have an improvement on VOC-MLT, but on COCO-MLT, adaptive class suppression leads to a decrease of mAP. Moreover, for adaptive class suppression, tuning is more difficult. The change of threshold has a great influence on the result. It is not steady. For negative-tolerant regularization, the effect from parameters is not too much. Therefore, negative-tolerant regularization is steadier.

5.4 Focal

Table 4 shows the comparison of two focal strategies. Obviously, the performance of asymmetric focal is better. This may be because two same focal parameters make traditional focal loss reduce the effect of both positive and negative classes, while asymmetric focal reserves effects of positive classes.

5.5 Swin-Transformer

Swin-Transformer causes a great improvement on both VOC-MLT and COCO-MLT. As can be seen from table 5, without any other methods which is designed for multi-label long-tailed classification, the performance of Swin-Transformer (Tiny) is still better than ResNet 50. With the expansion of its structure, its performance improves gradually and achieve 90.21%. This is because the feature extracted by Swin-Transformer is better than ResNet. This means a good feature extractor has a great effect on long-tailed classification problem.

Furthermore, according to table 6, these methods are still useful. On COCO-MLT, the improvement is obvious. Compared with baseline, there is a nearly 5% increase. However, on VOC-MLT, the improvement is only 1.23%. This might be because the generated features of

VOC-MLT is good. It is difficult to improve more.

5.6 Feature Decoupling Learning

Multi-layer perceptron can be used as a decoupler. The performance can achieve 55.85%. However, it has a disadvantage. Its space consumption is too large. Matrix decoupler alleviates this problem. Meanwhile, the performance is increased to 62.78%, but it is still not high. In addition, the model can converge by one-hot encoding representation, but by word-vector representation method, it cannot converge. This means word-vector could not be appropriate as class representation vectors.

6 Conclusion and Future Work

In this work, I compare different methods about multi-label long-tailed distribution problem. Current challenges of this problem are co-occurrence and dominance of negative labels. I conduct the experiments including augmentation strategies and 6 methods about these two challenges and compare their performance.

In conclusion, to some extent, distribution-balanced reweighting is helpful for co-occurrence problem, but not too much. Co-occurrence is still a great challenge for multi-label problem. If the relation among labels can be addressed better, the result may be improved more. Although negative-tolerant regularization and asymmetric focal seems perform well, parameter tuning is not easy. To find a high-performance adaptive approach may be more useful for industry.

In addition to these, I also conduct the experiments about feature extractor. The experiments shows that a good extractor has a great positive effect on the performance. Swin-Transformer is better than Resnet. However, its structure is larger. This causes a low running speed. In the future, to optimize its structure is able to be done.

At last, I proposed a feature decoupling learning strategy. It exists some disadvantages. Firstly, the performance of decoupler still can be improved. A large number of research can be done on this area. Furthermore, one-hot encoding representation is designed manually, and word-vector is from the relation among words. If class representation vectors are extracted from image features, the performance might be better. Consequently, in the future, some research about unsupervised and self-supervised learning can be conducted to summarize the feature of a set of images.

Reference

1. Hong, Y., Han, S., Choi, K., Seo, S., Kim, B. and Chang, B., 2021. Disentangling Label Distribution for Long-tailed Visual Recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 6626-6636).
2. Jamal, M.A., Brown, M., Yang, M.H., Wang, L. and Gong, B., 2020. Rethinking class-balanced methods for long-tailed visual recognition from a domain adaptation perspective. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 7610-7619).
3. Wang, X., Lyu, Y. and Jing, L., 2020. Deep generative model for robust imbalance classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 14124-14133).
4. Zhou, B., Cui, Q., Wei, X.S. and Chen, Z.M., 2020. 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).
5. Liu, Z., Luo, P., Wang, X. and Tang, X., 2015. Deep learning face attributes in the wild. In *Proceedings of the IEEE international conference on computer vision* (pp. 3730-3738).
6. Chen, T., Xu, M., Hui, X., Wu, H. and Lin, L., 2019. Learning semantic-specific graph representation for multi-label image recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (pp. 522-531).
7. Yazici, V.O., Gonzalez-Garcia, A., Ramisa, A., Twardowski, B. and Weijer, J.V.D., 2020. Orderless recurrent models for multi-label classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 13440-13449).
8. Chen, Z.M., Wei, X.S., Wang, P. and Guo, Y., 2019. Multi-label image recognition with graph convolutional networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 5177-5186).
9. Ridnik, T., Ben-Baruch, E., Zamir, N., Noy, A., Friedman, I., Protter, M. and Zelnik-Manor, L., 2021. Asymmetric Loss for Multi-Label Classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (pp. 82-91).
10. Kang, B., Xie, S., Rohrbach, M., Yan, Z., Gordo, A., Feng, J. and Kalantidis, Y., 2019. Decoupling representation and classifier for long-tailed recognition. *arXiv preprint arXiv:1910.09217*.
11. Cui, Y., Song, Y., Sun, C., Howard, A. and Belongie, S., 2018. Large scale fine-grained categorization and domain-specific transfer learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4109-4118).
12. Shen, L., Lin, Z. and Huang, Q., 2016, October. Relay backpropagation for effective learning of deep convolutional neural networks. In *European conference on computer vision* (pp. 467-482). Springer, Cham.
13. Guo, H., Zheng, K., Fan, X., Yu, H. and Wang, S., 2019. Visual attention consistency under image transforms for multi-label image classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 729-739).
14. Wang, P., Han, K., Wei, X.S., Zhang, L. and Wang, L., 2021. Contrastive Learning based Hybrid Networks for Long-Tailed Image Classification. In *Proceedings of the IEEE/CVF*

Conference on Computer Vision and Pattern Recognition (pp. 943-952).

15. Guo, H. and Wang, S., 2021. Long-Tailed Multi-Label Visual Recognition by Collaborative Training on Uniform and Re-balanced Samplings. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 15089-15098).
16. Wu, T., Huang, Q., Liu, Z., Wang, Y. and Lin, D., 2020, August. Distribution-balanced loss for multi-label classification in long-tailed datasets. In *European Conference on Computer Vision* (pp. 162-178). Springer, Cham.
17. Chen, B., Li, J., Lu, G., Yu, H. and Zhang, D., 2020. Label co-occurrence learning with graph convolutional networks for multi-label chest x-ray image classification. *IEEE journal of biomedical and health informatics*, *24*(8), pp.2292-2302.
18. Wang, T., Zhu, Y., Zhao, C., Zeng, W., Wang, J. and Tang, M., 2021. Adaptive Class Suppression Loss for Long-Tail Object Detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 3103-3112).
19. Lin, T.Y., Goyal, P., Girshick, R., He, K. and Dollár, P., 2017. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision* (pp. 2980-2988).
20. Krizhevsky, A., Sutskever, I. and Hinton, G.E., 2017. ImageNet classification with deep convolutional neural networks. *Communications of the ACM*, *60*(6), pp.84-90.
21. He, K., Zhang, X., Ren, S. and Sun, J., 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770-778).
22. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł. and Polosukhin, I., 2017. Attention is all you need. In *Advances in neural information processing systems* (pp. 5998-6008).
23. Zhang, Z., Zhang, H., Zhao, L., Chen, T. and Pfister, T., 2021. Aggregating nested transformers. *arXiv preprint arXiv:2105.12723*.
24. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S. and Guo, B., 2021. Swin transformer: Hierarchical vision transformer using shifted windows. *arXiv preprint arXiv:2103.14030*.
25. Everingham, M., Eslami, S.A., Van Gool, L., Williams, C.K., Winn, J. and Zisserman, A., 2015. The pascal visual object classes challenge: A retrospective. *International journal of computer vision*, *111*(1), pp.98-136.
26. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P. and Zitnick, C.L., 2014, September. Microsoft coco: Common objects in context. In *European conference on computer vision* (pp. 740-755). Springer, Cham.