https://doi.org/10.2352/EI.2022.34.11.HVEI-189

INDeeD: Identical and Disparate Feature Decomposition from Multi-label Data

Tserendorj Adiya and Seungkyu Lee
Department of Computer Science and Engineering, Kyung Hee University, Republic of KOREA

Abstract

Deep learning technology has made a significant improvement in image recognition performance. Unlike single-labeled training and inference, multi-labeled classification tasks hardly characterize individual label in the training of deep neural networks due to co-occurrence of the labels. Training data contains few samples of separated single label and the networks learn diverse compositions of labels from the data. Contextual bias caused by the co-occurrence of labels disturbs multi-label classification. We propose Identical and Disparate Feature Decomposition (INDeeD) from multi-label data that explicitly learn the characteristics of individual label. By training a backbone network combined with Identical and Disparate blocks on the instance pairs of partially common and contrastable labels, the network is generalized to decompose and learn individual label features. Proposed INDeeD scheme can be simply incorporated in any type of networks. We use ML-MNIST, ML-CIFAR-10, VOC-2007, and MS-COCO datasets to evaluate the performance of INDeeD showing improved mAP over baseline.

Introduction

Recent advances in deep learning approaches have demonstrated excellent performance gain in image based recognition tasks. Advanced networks [9, 23, 26, 29] significantly improve single-label classification performance being trained on large scale data set such as ImageNet [24]. Recent challenging classification tasks with compositive and congregated class types require complex training data sets of larger data size, higher image resolution, multiple modalities, and complex structure of semantic visual components with multiple labels. For example, object detection methods [20, 30, 22, 28] are evaluated on data sets [18, 8, 15]. High-resolution images of the data sets contain multiple class objects with respective bounding boxes on complex backgrounds with mixed situations. Unlike single-label instance classification, however, in the classification task with multi-label instance requires comprehensive and contextual understanding of the existence of each label. In general there is no explicit guide for multi label in an instance such as a bounding box in object detection training. These situations make it difficult to learn features for precise local locations. In addition, the higher frequency of the label co-occurrence, the more confusing it is in learning to accurately represent the characteristics of individual categories. For example in Figure 1 (a), person frequently co-occurred with motorcycle or horse in single instance corresponding "a person riding a motorcycle" or "a person riding a horse" respectively. With such instances networks learn comprehensive features of such multi-label cases rather than learning individual object. In other words, when an object to be recognized is co-occured with other objects or appears in a specific environment in most cases, networks tend to have contextual bias and the characteristics of each object cannot be appropriately expressed. Recently, with the success of graph convolutional network (GCN)[12], GCN based methods [4, 5, 31, 33] have introduced in multi-label tasks. Chen *et al.*[5] use the frequency of instance co-occurrence as training information. They create a graph structure with each class vector obtained by training to learn the semantic relationship between classes using conditional probability. Besides, there have been several studies that tackle the problem in the perspective of class imbalance. Wu *et al.*[32] have made efforts to mitigate data imbalance of relatively less appeared labels in multi-label samples. They propose a re-sampling method to make class distribution uniform. However, even though they increase the number of samples by re-sampling for the classes of multi-label [3, 2, 10, 25], new samples with still co-occurring labels does not resolve the training problem of each label. Ben-Baruch *et al.*[1] propose a loss function to assign different weights to different labels so that the instance distribution of each label becomes uniform. Although basic principle is similar to Focal Loss [17], they achieve meaningful performance improvement by reducing the dominance of less appeared labels. However, adapting the loss does not directly mitigate the contextual bias. In addition, Ben-Baruch *et al.*[1] experimented in an environment pre-trained with ImageNet [24], and when training with large data such as ImageNet, contextual bias is less appeared.

In this work, we introduce a novel training method on multi-label data, proposing identical and disparate feature decomposition (INDeeD) to alleviate contextual bias for individual label even in online learning situation. INDeeD training scheme generalizes a backbone network to learn features for individual label explicitly via identical and disparate blocks. Identical and disparate blocks extract individual class features through similarity-based masking training on instance pairs of both sharing and non-conflict labels named "partial-coco pairs". The detailed definition is in the section . For example, suppose that the "partial-coco pair" images X_a and X_b contain {person, motorcycle} and {person, horse}, respectively. In order to decompose the features corresponding to each class in the two samples, the similar and dissimilar parts in the two images are to be decomposed. So, the Identical block extracts features with high similarity (features corresponding to person) from X_a and X_b, and the disparate block extracts features that are not similar (features corresponding to motorcycle or horse) from X_a and X_b. Extracting features for individual labels in this way makes the classifier more robust providing augmented training sample for individual labels in the feature space. INDeeD training is a kind of constraint alleviating the contextual bias described above. In other words, it restricts the parts with high similarity to express the same class in the "partial-coco

pairs" image features and force the features with low similarity to express different classes.

The contributions of our work includes: 1) a novel INDeeD training method with Indentical and Disparate blocks that decompose individual label representation from multi-label data 2) Two types of multi-label toy sets (ML-MNIST and ML-Cifar-10) for analytical evaluation for multi-label classification tasks and 3) Extensive evaluation on 8 variations of our own toy sets, VOC-2007 and MS-COCO data sets.

Method

Figure 1 illustrates overall structural flow of our INDeeD training. X_a and X_b are multi-labeled input images with respective label set $\mathbf{y}_a, \mathbf{y}_b \subset \mathbf{Y}$, where $\mathbf{y} = \{y^1, y^2, ..., y^c\}$ is all class labels and c is total number of class labels in data set. $f_a, f_b \in \mathbb{R}^{1\times n}$ are n-dimensional feature vectors of images X_a and X_b in the latent space where identical and disparate blocks are operated. In order to learn individual class representation from multi-label data, our INDeeD need to partial-coco pair. Partial-coco pair (X_a, X_b) is randomly chosen from all samples of mini-batch that satisfy:

$$(\mathbf{y}_a \cap \mathbf{y}_b) \neq \varnothing \tag{1}$$

$$((\mathbf{y}_a - \mathbf{y}_b) \cup (\mathbf{y}_b - \mathbf{y}_a)) \neq \varnothing \tag{2}$$

If there is no remaining such pair in mini-batch, default pair (X_a, X_a) is used.

Identical and Disparate Blocks

Feature vectors $f_a, f_b \in \mathbb{R}^{1\times n}$ of chosen image pairs (X_a, X_b) are fed to identical and disparate block respectively. Detailed structure of identical, disparate block is illustrated in Figure 1(b). To represent individual class feature, we need to define the identical map m for the extraction of identical features as follows.

$$m = \frac{f_a \otimes f_b}{\| f_a \otimes f_b \|_2} \tag{3}$$

where operator $\otimes$ indicates element-wise multiplication and $\| \cdot \|_2$ is L2 normalization. Alternative optional normalization is min-max defined as follows.

$$m = \frac{(f_a \otimes f_b) - \min(f_a \otimes f_b)}{\max(f_a \otimes f_b) - \min(f_a \otimes f_b)}, \tag{4}$$

$\min(\cdot)$ returns minimum value and $\max(\cdot)$ returns maximum value from input vector. The reason of using two different normalization, the normalized identical map attributes are effects bit different way to learn individual class. L2 normalization generally produces a smooth identical map, whereas min-max normalization produces relatively sharp map. These differences in attributes affect training. The influence of the min-max normalization method on the feature space has a more whirlwind influence than L2. However, min-max normalization is more sensitive to outliers than L2. So, you can decide which normalization to use depending on the quality of the data. Since f_a,f_b is a value that has passed through an activation function such as ReLU, no matter which norm method is used, m has a consisting of values between 0 and 1.

Identical map $m \in \mathbb{R}^{1\times n}$ produced by (3) or (4) that expresses how similar each element of f_a and f_b. Similar common feature elements of f_a and f_b are boosted and other elements are suppressed in two outputs $f_{a\cap b} = m \otimes f_a$, $f_{b\cap a} = m \otimes f_b$ of identical block. The features $f_{a\cap b}$ and $f_{b\cap a}$ obtained by the identical block are explicitly represent to the common category in the partial-coco pair X_a and X_b. For this reason, the labels $\mathbf{y}_a \cap \mathbf{y}_b$ is assigned to f_a and f_b.

We have previously defined the condition for partial-coco pair as in (1) and (2). Identical block can be executed even if only condition (1) is satisfied. However, the Disparate block can extract meaningful features only when both conditions (1) and (2) are satisfied. The Disparate block extracts features corresponding to different categories from the partial-coco pair, which has the opposite role of the Identical block. Therefore, $(1-m)$, which is the opposite map of the identical map, is used: $f_{a-b} = (1-m) \otimes f_a$, $f_{b-a} = (1-m) \otimes f_b$. A label $\mathbf{y}_a - \mathbf{y}_b$ is assigned to the feature f_{a-b} extracted in this way, and a label $\mathbf{y}_b - \mathbf{y}_a$ is assigned to f_{b-a}. Note that f_{a-b} and f_{b-a} are completely different feature, because those two feature represents completely different categories. Finally, Identical and Disparate blocks add four new feature vectors building subdivided descriptions on the class labels (feature vectors $F(X_a, X_b)$ below are fed to following classifier).

$$F(X_a, X_b) = \{f_a, f_b, f_{a\cap b}, f_{b\cap a}, f_{a-b}, f_{b-a}\} \tag{5}$$

Objective Function

Loss function of our classification is defined as follows.

$$L = L_O + \lambda(L_I + L_S) \tag{6}$$

where loss terms L_O, L_I, and L_S are calculated from classification based on original features (f_a, f_b), identical features $(f_{a\cap b}, f_{b\cap a})$, and disparate features (f_{a-b}, f_{b-a}) respectively.

$$L_O = BCE(p_a, \mathbf{y}_a) + BCE(p_b, \mathbf{y}_b) \tag{7}$$

$$L_I = BCE(p_{a\cap b}, \mathbf{y}_{a\cap b}) + BCE(p_{b\cap a}, \mathbf{y}_{b\cap a}) \tag{8}$$

$$L_S = BCE(p_{a-b}, \mathbf{y}_{a-b}) + BCE(p_{b-a}, \mathbf{y}_{b-a}) \tag{9}$$

where logits p_a, p_b, $p_{a\cap b}$, $p_{b\cap a}$, p_{a-b}, p_{b-a} are from corresponding feature f_a, f_b, $f_{a\cap b}$, $f_{b\cap a}$, f_{a-b}, f_{b-a} and binary cross-entropy(BCE) of a classifier is defined as follows.

$$BCE(f, l) = -\sum_i l_i \log(\sigma(p_i)) + (1-l_i)\log(1-\sigma(p_i)) \tag{10}$$

with using sigmoid $\sigma(x) = (1+\exp(x))^{-1}$. Vector p contains classifier logits and l denotes the vector of binary labels, and i is class index. In this classification, based on the (feature, corresponding label) $(f_a, \mathbf{y}_a)$ and $(f_a, \mathbf{y}_a)$, new labels $\mathbf{y}_{a\cap b}$, $\mathbf{y}_{b\cap a}$, $\mathbf{y}_{a-b}$, $\mathbf{y}_{b-a}$ for new four feature vectors $f_{a\cap b}$, $f_{b\cap a}$, f_{a-b}, f_{b-a} are defined as follows.

$$\begin{aligned} \mathbf{y}_{a\cap b} &= \mathbf{y}_{b\cap a} = \mathbf{y}_a \cap \mathbf{y}_b \\ \mathbf{y}_{a-b} &= \mathbf{y}_a - \mathbf{y}_b = \mathbf{y}_a \cap \mathbf{y}_b \\ \mathbf{y}_{b-a} &= \mathbf{y}_b - \mathbf{y}_a = \mathbf{y}_b \cap \mathbf{y}_a \end{aligned} \tag{11}$$

INDeeD leverages backbone networks to extract f_a, f_b features good to extract following identical features $(f_{a\cap b}, f_{b\cap a})$ and disparate features (f_{a-b}, f_{b-a}).

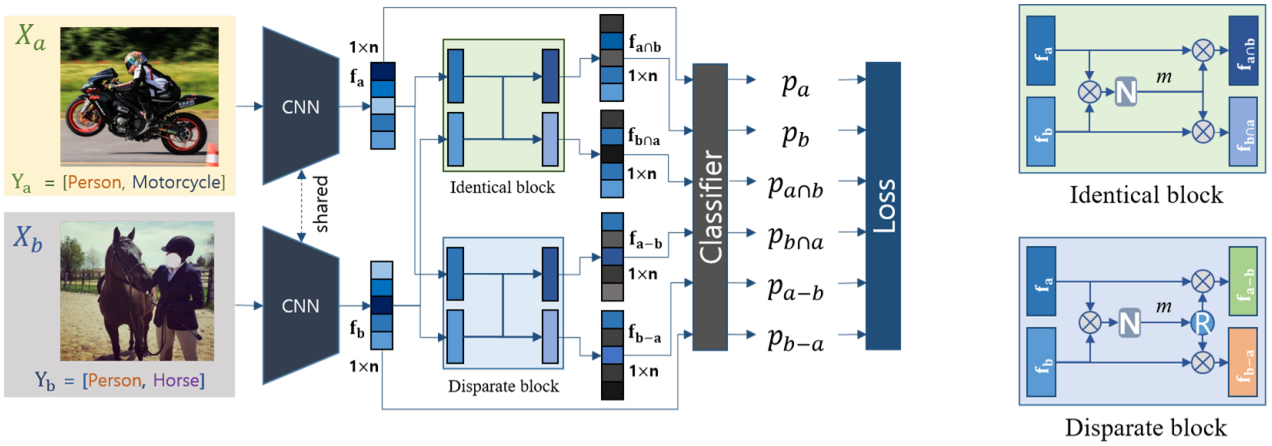


Figure 1: (a) Overall diagram of proposed method: $f_{a\cap b}$, $f_{b\cap a}$, f_{a-b} and f_{b-a} features are extracted by Identical and Disparate blocks from features f_a and f_b of input images X_a, X_b. Classification is conducted based on the six feature vectors. (b) Detailed structure of Identical and Disparate blocks: $\otimes$ stands for element-wise product and **N** indicates normalization. L2 norm or min-max is used as **N**. **R** indicates (1-m).

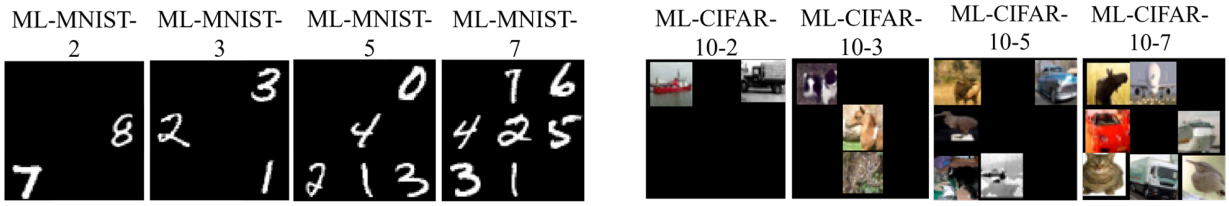


Figure 2: Sample images of ML-MNIST and ML-CIFAR-10 data sets cunstructed for multi-label classification evaluation: Number of labels are 2, 3, 5, and 7.

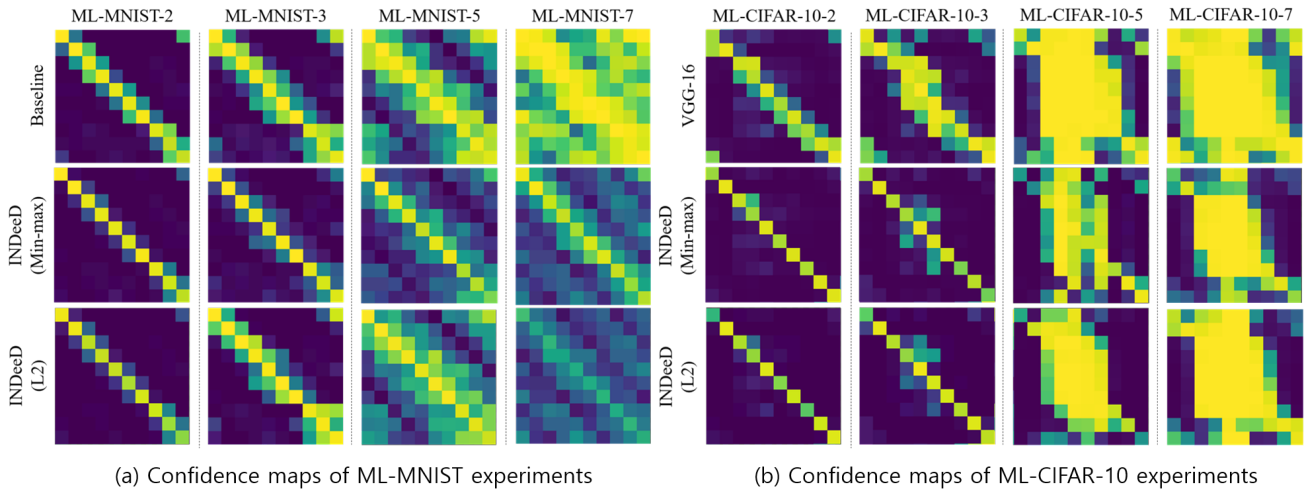


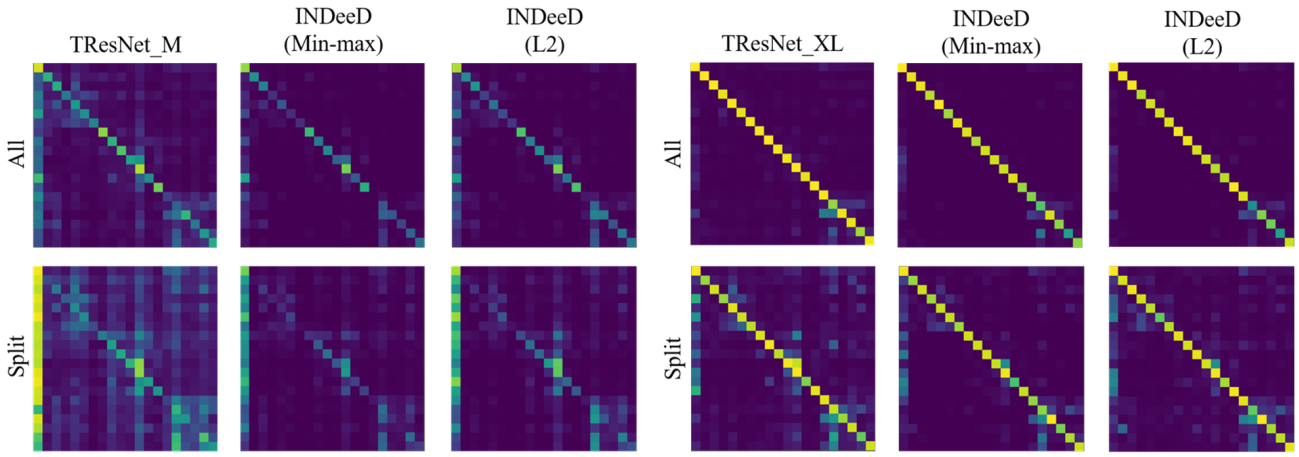
Figure 3: Confidence maps of the tests on ML-MNIST and ML-CIFAR-10 data sets: Vertical axis of confidence map corresponds to label index of test data. Horizontal-axis is average output probability value of corresponding test instances of vertical-axis.

Experimental Evaluation

We evaluate proposed method on MNIST[16], CIFAR-10[13], PASCAL VOC-2007[8] and MS-COCO[19] data sets. MNIST [16] and CIFAR-10 [13] are configured for single label classification task. For an analytic evaluation, we con-

ML-MNIST-N	Methods	zero	one	two	three	four	five	six	seven	eight	nine	**mAP**
N=2	Baseline	99.67	99.29	95.34	97.01	97.70	95.47	99.05	98.08	84.92	88.66	94.04
	INDeeD (L2)	99.81	99.70	97.43	98.24	98.67	97.64	98.99	**98.86**	93.13	93.87	97.66
	INDeeD (Min-max)	**99.82**	**99.78**	**98.02**	**98.42**	**98.72**	**97.91**	**99.39**	98.73	**94.69**	**94.07**	**98.38**
N=3	Baseline	99.13	99.63	97.14	94.20	93.29	92.76	99.08	95.20	74.02	88.83	92.47
	INDeeD (L2)	99.68	99.73	98.09	97.13	96.28	98.62	99.06	96.54	92.50	94.97	97.72
	INDeeD (Min-max)	**99.80**	**99.91**	**99.26**	**98.05**	**97.47**	**99.12**	**99.44**	**97.78**	**96.83**	**96.62**	**98.31**
N=5	Baseline	99.34	97.85	96.38	96.37	93.99	91.56	97.57	88.61	87.29	60.24	86.40
	INDeeD (L2)	**99.70**	**99.51**	**97.88**	**97.00**	**96.41**	**98.56**	**98.47**	**96.66**	**92.55**	**94.71**	**95.14**
	INDeeD (Min-max)	99.70	99.35	97.79	95.87	96.37	96.97	97.95	92.79	92.18	66.20	94.56
N=7	Baseline	97.70	95.85	78.52	75.72	52.48	84.30	92.15	86.76	57.38	67.61	78.84
	INDeeD (L2)	99.47	99.79	94.90	95.22	96.07	**98.17**	99.34	**98.43**	**98.39**	**96.07**	**97.73**
	INDeeD (Min-max)	**99.76**	**99.84**	**96.53**	**96.34**	**97.84**	98.05	**99.49**	97.96	95.75	95.90	97.69

Table 1: ML-MNIST test results of Baseline(simple CNN) and INDeeD normalized by min-max and L2 and evaluated by average precision (AP) and mean of AP (mAP)



(a) Confidence maps of TResNet-M trained on VOC-2007 dataset

(b) Confidence maps of TResNet-XL (pre-trained on ImageNet) trained on VOC-2007 dataset

Figure 4: Confidence maps of the tests on VOC-2007 *All* and *Split* data sets

ML-CIFAR-10-N	Methods	plane	car	bird	cat	deer	dog	frog	horse	ship	truck	**mAP**
N=2	Baseline	**97.59**	91.03	77.57	**93.38**	85.95	**91.42**	93.16	94.98	95.17	88.66	89.64
	INDeeD (L2)	92.00	97.86	**92.36**	79.33	94.11	91.25	96.78	95.21	97.59	96.06	93.23
	INDeeD (Min-max)	93.51	**97.96**	92.35	81.37	**95.14**	89.35	**96.81**	**96.77**	**96.71**	**96.61**	**93.67**
N=3	Baseline	90.48	**98.66**	91.33	81.92	**92.04**	74.19	94.66	93.96	**97.33**	93.57	90.81
	INDeeD (L2)	88.88	98.60	**93.72**	**85.34**	89.82	**89.73**	96.78	95.05	97.01	96.44	93.14
	INDeeD (Min-max)	**92.54**	98.65	92.75	83.81	91.83	89.41	**96.98**	**95.83**	97.27	**96.46**	**93.61**
N=5	Baseline	92.52	98.00	64.28	31.02	77.51	83.67	**32.25**	94.70	94.48	95.76	76.61
	INDeeD (L2)	89.18	**98.21**	69.88	65.10	**85.26**	71.15	29.06	95.38	**97.24**	95.83	79.63
	INDeeD (Min-max)	**94.09**	97.45	**78.73**	**67.88**	84.79	**83.84**	30.16	**97.08**	97.04	**96.59**	**82.55**
N=7	Baseline	**82.57**	98.06	39.17	23.12	19.42	17.02	48.96	94.37	96.49	85.02	60.73
	INDeeD (L2)	30.56	96.92	79.49	**68.71**	**61.45**	16.63	29.62	94.49	96.32	**96.57**	67.07
	INDeeD (Min-max)	80.86	**98.21**	**89.99**	35.25	20.72	**19.58**	**84.84**	**94.57**	**97.33**	90.89	**71.22**

Table 2: ML-CIFAR-10 test results of Baseline(VGG-16) and INDeeD normalized by min-max and L2 and evaluated by average precision (AP) and mean of AP (mAP)

struct multi-label MNIST (ML-MNIST) and multi-label CIFAR-10 (ML-CIFAR-10) data sets. We make new composite images (of size $3H \times 3W$) including 2, 3, 5, and 7 MNIST or CIFAR-10 images (of size $H \times W$) of different labels as shown in Figure 2. These new multi-label data sets enable us to control the number and composition of multi-label in each sample. VGG-16 [26], ResNet-50 [9], TResNet [23], and simple convolutional Neural Nets are used as backbone networks. For fair comparison between the baseline and proposed method, all experiments are performed in the same setup. Since INDeeD aims to generalize a backbone network, Indentical and Disparate blocks are not used in the inference step.

Evaluation on ML-MNIST

MNIST is handwritten image data set that consists of 60,000 training and 10,000 test samples divided into 10 digit categories. Image Size of MNIST is 28×28 and ML-MNIST image size with multiple labels is 84×84. ML-MNIST-2, ML-MNIST-3, ML-

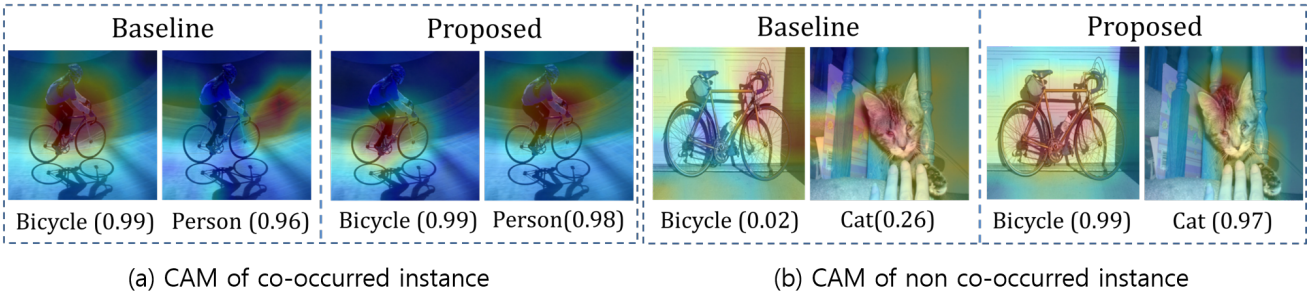


Figure 5: Class activation maps of INDeeD and Baseline: INDeeD CAM shows better localization and higher confidence scores.

Method	Scratch				Pre-trained			
	All		Split		All		Split	
	single	entire	single	entire	single	entier	single	entire
Baseline	54.19	58.34	29.73	39.90	95.15	**94.59**	88.89	89.11
INDeeD (min-max)	55.23	59.66	29.68	39.81	**95.29**	94.46	91.40	89.41
INDeeD (L2)	**55.76**	**60.04**	**29.77**	**40.71**	95.21	94.37	**91.64**	**90.60**

Table 3: VOC-2007 test results of scratch and pre-trained (on ImageNet): Evaluation metric is mean average precision(in %). 'Scratch' baseline is TResNet-M and 'Pre-trained' baseline is TResNet-XL.

MNIST-5, and ML-MNIST-7 are multi-label MNIST with 2, 3, 5, and 7 multiple labels in one instance respectively (Figure 2 (a)). With ML-MNIST we investigate robustness and varying performance of INDeeD along the increasing complexity of multiple labels. Simple CNN[14] with three convolutional layers and two fully-connected layers is baseline network. Adam Optimizer [11] with learning rate of 0.001 and reduced learning rate by 0.5 for each epoch is used and trained for 30 epochs. Test data consists of images with single class label.

Figure 3(a) shows average confidence score maps of single class classification test trained on various ML-MNIST data sets. Vertical-axis of confidence map represents each single class of test data. Horizontal-axis is average output probability value of corresponding test instances of vertical-axis. Baseline network shows seriously decreasing classification performance as the number of multi-label increases. Confidence maps in the second and third rows of proposed INDeeD with L2 and min-max normalization, respectively. Both min-max and L2 normalization cases show clearly improved average probability over baseline. Probability of correct label class (on the diagonal line of the confidence map) becomes more distinct in the map. Especially in min-max normalization case with the most complex multi-label case (ML-MNIST-7), all of test classes show best and outstanding test probability in correct label class (on the diagonal line of the confidence map) compared to others. Table 1 compares mean average precision (mAP) of the tests. Proposed method outperforms baseline in most cases. Performance gain of our method over baseline in mAP increases as the number of multi-label increases (from 3.98 to 18.85 in Min-max case). Furthermore, mAP of our proposed method is stable regardless of the number of multi-label. Over all in the test, min-max normalization represents an individual class more distinctly than L2 normalization.

Evaluation on ML-CIFAR-10

CIFAR-10 [13] consists of 60,000 color images of 10 classes (6,000 images per class). 50,000 images are for training and 10,000 images are for test. Original CIFAR-10 image size is 32×32 and ML-CIFAR-10 image size with various number of multiple labels becomes 96×96. ML-CIFAR-10-2, ML-CIFAR-10-3, ML-CIFAR-10-5, and ML-CIFAR-10-7 are multi-label CIFAR-10 data sets with 2, 3, 5, and 7 multiple labels in one instance, respectively (Figure 2 (b)). We use VGG-16 [26] as a backbone network and Adam [11] as an optimizer. Learning rate is 0.001. Baseline method and INDeeD are trained for 300 epochs and learning rate is adjusted to SGDR [21] and used well known augmentation [7, 6] methods. Figure 3 (a) shows confidence maps of single class classification test results trained on various ML-CIFAR-10 data sets. Proposed INDeeD (in the second and third rows) shows improved classification results over the baseline. Compared to ML-MNIST, ML-CIFAR-10 with real object images show limited single class classification performance as the number of multi-label increases. This tendency is more obvious in mAP scores shown in Table 2. In all cases, our proposed method shows mAP gains(4.02%, 2.79%, 5.94%, and 10.49% improvement). Similar to ML-MNIST, ML-CIFAR-10 also shows better results with min-max normalization than L2 norm. This is because that ML-MNIST and ML-CIFAR-10 are composed of relatively clean images of multiple objects in line located in predetermined locations.

Evaluation on VOC-2007 and MS-COCO

Pascal Visual Object Classes Challenge (VOC 2007) [8] for multi-label recognition contains images of 20 object categories. Each image falls in around 2.5 categories in average. Pascal-VOC is divided into 5,011 trainval images and 4,952 test images. We define two training schemes: *All* and *Split*. *All* uses entire trainval set (5,011 images). *Split* uses multi-label images only (2208 images). *Split* training verifies the effectiveness of individual class classification purely only on multi-label Class Data. We define two test schemes: *single* and *entire*. *single* scheme tests only on single labeled instances (2848 images). *entire* scheme uses entire test data of VOC-2007 (4,952 images). We use TResNet [23] M and XL sizes as backbone networks. All experiments conducted by TResNet-M begin with learning rate of 0.001 on Adam opti-

Block	Normalization	ML-MNIST-N				ML-CIFAR-10-N			
		N=2	N=3	N=5	N=7	N=2	N=3	N=5	N=7
Idenctical	Min-max	96.73	97.52	93.34	90.42	92.51	94.12	83.51	57.17
	L2	97.19	95.55	90.15	90.03	92.59	93.29	77.12	52.74
Disparate	Min-max	97.78	97.76	91.75	94.07	93.13	92.60	84.92	73.93
	L2	97.88	97.64	94.09	88.58	92.05	91.42	67.90	63.69

Table 4: Ablation Study: mAP(in %) with either identical or disparate block on ML-MNIST and ML-CIFAR-10

mizer. Learning rate is adjusted with OneCycleLR [27] scheduler for 400 epochs.

Figure 4 shows confidence maps of TResNet-M and TResNet-XL compared with INDeeD results normalized by min-max and L2 normalization using both *All* and *Split* training schemes. INDeeD improves the class separation in confidence maps for both networks over baseline. Table 3 "Scratch" shows that INDeeD (normalized by L2) shows 1.7% higher mAP for *All* scheme (1.57% higher for single-label tests) and 0.81% higher for *Split* scheme (0.04% higher for single-label tests) than baseline. In VOC-2007, L2 norm shows higher mAPs than min-max. In general, L2 norm is less sensitive to out-liers than min-max. VOC-2007 is composed of natural images of high resolution with irregular object positions with increased variations in many aspects. Overall performance gain of VOC-2007 is smaller than ML-MNIST and ML-CIFAR-10. Yet INDeeD efficiently learns individual label. We also conduct experiment on pre-trained on ImageNet. In this experiment we use TResNet-XL as baseline. Each experiment trained for 80 epoch and over all setups are same as TResNet-M experiment. Confidence maps and mAP scores are shown in in Figure 4(b) and Figure 4(b) respectively. Using *entire* training data, there is no significant performance differences between baseline and INDeeD. However, INDeeD show mAP gain in *Split* data (see Table 3 "Pre-trained" 1.49% in *entire* , 2.75% in *single*). The reason is that the first class (person class) confidence score is much higher than the correct class in Figure 4(b) *Split*. On the other side first class(person class) confidence score is reduced in Figure 4(b) *Split*. In other words, pre-training alleviates the contextual bias of the first class (person class). Excessive growth of other class confidence scores than the correct class confidence score is prevents the effective of extract meaningful identical and disparate features. It means that INDeeD alleviates contextual bias in high co-occurred situation.

We test the effectiveness of INDeeD method on the large scale dataset MS-COCO. MS-COCO is primarily built for object recognition in the context of scene understanding. Training set is composed of 82,783 images containing common objects in the scenes. The objects are categorized into 80 classes with about 2.9 object labels per image. Since the ground-truth labels of test set is not available, we evaluate all methods on the validation set (40,504 images). We use ResNet-50 as baseline trained 200 epochs with Adam optimizer and learning rate of 0.001 (adjusted with OneCycleLR scheduler). Similar to VOC-2007, L2 norm shows a higher mAP (71.56 for Single and 71.20 for Entire) than min-max (71.01 for Single and 70.15 for Entire) and baseline (69.21 for Single and 69.11 for Entire).

Ablation Study

Identical and disparate blocks are added separately to our experiments on ML-MNIST, ML-CIFAR-10, and VOC-2007 data sets to analyze the effects of identical and disparate blocks, respectively. With ML-MNIST data sets, min-max shows better performance than L2 by average 1.19% for identical and 0.78% for disparate, respectively (Table 4 ML-CIFAR-10). Also with ML-CIFAR-10 data sets, min-max shows better performance than L2 by average 1.89% for identical and 7.13% for disparate, respectively (Table 4 ML-MNIST).

Methods	Normalization	All		Split	
		single	entire	single	entire
Identical	Min-max	54.21	58.01	27.47	38.09
	L2	56.53	59.29	28.91	40.13
Disparate	Min-max	53.53	56.79	27.98	38.69
	L2	44.31	51.01	28.30	38.98

Table 5: Ablation Study: mAP(in %) with either identical or disparate block on VOC-2007

In VOC-2007 dataset, the identical block with L2 normalization shows 1.66% better performance than min-max normalization and the disparate block with min-max normalization shows 2.75% better performance than L2 normalization (see Table 5).

We compare not only mAP but also how much INDeeD alleviates contextual bias through visualization of class activation map (CAM). Looking at the baseline CAM in Figure 5(a), the CAM corresponding to "bicycle" covers both "bicycle" and "person" area, and the cam corresponding to "person" is activated in the background(not related person). That is, the baseline method is learned to express comprehensively focusing on the characteristics of people, bicycles and backgrounds rather than individual characteristics of people and bicycles. As a result contexts are biased. However, look at the results of training with the INDeeD method in Figure 5(a), the areas corresponding to "bicycle" and "people" are precisely activated. In addition to this, look at Figure 5(b), you can see that the baseline method extracts inaccurate features when there is non co-occurring category. However, it can be seen that our method accurately represents the individual features even for non-co-occurred samples.

Conclusion

In this paper, we introduced INDeeD, a training method that alleviates the problem of contextual bias caused by co-occurrence. Through the experimental results conducted on the ML-MNIST, ML-CIFAR-10, VOC-2007, and MS-COCO data sets, it was confirmed that our method is effective in alleviating contextual bias. However, as the data diversity and pattern complexity increase, the performance increase decreases, but it is still effective. We leave these issues as a task to be addressed in the future.

References

[1] Emanuel Ben-Baruch et al. *Asymmetric Loss For Multi-Label Classification*. 2020. arXiv: 2009.14119 [cs.CV].

[2] Mateusz Buda, Atsuto Maki, and Maciej A Mazurowski. "A systematic study of the class imbalance problem in convolutional neural networks". In: *Neural Networks* 106 (2018), pp. 249–259.

[3] Jonathon Byrd and Zachary Lipton. "What is the effect of importance weighting in deep learning?" In: *International Conference on Machine Learning*. PMLR. 2019, pp. 872–881.

[4] Tianshui Chen et al. "Learning semantic-specific graph representation for multi-label image recognition". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019, pp. 522–531.

[5] Zhao-Min Chen et al. "Multi-label image recognition with graph convolutional networks". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 5177–5186.

[6] Ekin D Cubuk et al. "Randaugment: Practical automated data augmentation with a reduced search space". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*. 2020, pp. 702–703.

[7] Terrance DeVries and Graham W Taylor. "Improved regularization of convolutional neural networks with cutout". In: *arXiv preprint arXiv:1708.04552* (2017).

[8] Mark Everingham et al. "The pascal visual object classes (voc) challenge". In: *International journal of computer vision* 88.2 (2010), pp. 303–338.

[9] Kaiming He et al. "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[10] Nathalie Japkowicz and Shaju Stephen. "The class imbalance problem: A systematic study". In: *Intelligent data analysis* 6.5 (2002), pp. 429–449.

[11] Diederik P Kingma and Jimmy Ba. "Adam: A method for stochastic optimization". In: *arXiv preprint arXiv:1412.6980* (2014).

[12] Thomas N. Kipf and Max Welling. *Semi-Supervised Classification with Graph Convolutional Networks*. 2017. arXiv: 1609.02907 [cs.LG].

[13] Alex Krizhevsky, Geoffrey Hinton, et al. "Learning multiple layers of features from tiny images". In: (2009).

[14] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "Imagenet classification with deep convolutional neural networks". In: *Advances in neural information processing systems* 25 (2012), pp. 1097–1105.

[15] Alina Kuznetsova et al. "The Open Images Dataset V4". In: *International Journal of Computer Vision* 128.7 (Mar. 2020), pp. 1956–1981. ISSN: 1573-1405. DOI: 10.1007/s11263-020-01316-z. URL: http://dx.doi.org/10.1007/s11263-020-01316-z.

[16] Yann LeCun et al. "Gradient-based learning applied to document recognition". In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324.

[17] Tsung-Yi Lin et al. "Focal loss for dense object detection". In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 2980–2988.

[18] Tsung-Yi Lin et al. *Microsoft COCO: Common Objects in Context*. 2015. arXiv: 1405.0312 [cs.CV].

[19] Tsung-Yi Lin et al. "Microsoft coco: Common objects in context". In: *European conference on computer vision*. Springer. 2014, pp. 740–755.

[20] Ze Liu et al. "Swin transformer: Hierarchical vision transformer using shifted windows". In: *arXiv preprint arXiv:2103.14030* (2021).

[21] Ilya Loshchilov and Frank Hutter. "Sgdr: Stochastic gradient descent with warm restarts". In: *arXiv preprint arXiv:1608.03983* (2016).

[22] Yoshitomo Matsubara. "torchdistill: A Modular, Configuration-Driven Framework for Knowledge Distillation". In: *International Workshop on Reproducible Research in Pattern Recognition*. Springer. 2021, pp. 24–44.

[23] Tal Ridnik et al. "TResNet: High Performance GPU-Dedicated Architecture". In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. Jan. 2021, pp. 1400–1409.

[24] Olga Russakovsky et al. "Imagenet large scale visual recognition challenge". In: *International journal of computer vision* 115.3 (2015), pp. 211–252.

[25] Li Shen, Zhouchen Lin, and Qingming Huang. "Relay backpropagation for effective learning of deep convolutional neural networks". In: *European conference on computer vision*. Springer. 2016, pp. 467–482.

[26] Karen Simonyan and Andrew Zisserman. "Very Deep Convolutional Networks for Large-Scale Image Recognition". In: *3rd International Conference on Learning Representations, ICLR*. 2015. URL: http://arxiv.org/abs/1409.1556.

[27] Leslie N Smith and Nicholay Topin. "Super-convergence: Very fast training of neural networks using large learning rates". In: *Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications*. Vol. 11006. International Society for Optics and Photonics. 2019, p. 1100612.

[28] Chen Sun et al. "Revisiting unreasonable effectiveness of data in deep learning era". In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 843–852.

[29] Christian Szegedy et al. "Going deeper with convolutions". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 1–9.

[30] CY Wang, A Bochkovskiy, and HYM Liao. "Scaled-YOLOv4: Scaling Cross Stage Partial Network. arXiv 2020". In: *arXiv preprint arXiv:2011.08036* ().

[31] Ya Wang et al. "Multi-label classification with label graph superimposing". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 34. 07. 2020, pp. 12265–12272.

[32] Tong Wu et al. “Distribution-balanced loss for multi-label classification in long-tailed datasets”. In: *European Conference on Computer Vision*. Springer. 2020, pp. 162–178.

[33] Renchun You et al. “Cross-modality attention with semantic graph embedding for multi-label classification”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 34. 07. 2020, pp. 12709–12716.

Author Biography

TSERENDORJ ADIYA received the B.S. degree in computer engineering from Kyung Hee University, Yongin, South Korea, in 2019. His research interests include image classification, action recognition, computer vision, and machine learning.

SEUNGKYU LEE received the B.S. and M.S. degrees in electrical engineering from the Korea Advanced Institute of Science and Technology, Daejeon, South Korea, in 1997 and 1999, respectively, and the Ph.D. degree in computer science and engineering from The Pennsylvania State University in 2009. He has been a Research Engineer with the Technical Research Institute, Korea Broadcasting System, Seoul, South Korea, where he has been involved in the research on high-definition image processing, MPEG4 advanced video coding, and the standardization of terrestrial-digital mobile broadcasting. He has also been a Principal Research Scientist with the Advanced Media Laboratory, Samsung Advanced Institute of Technology, Yongin, South Korea. He is currently an Associate Professor with Kyung Hee University. His research interests include time-of-flight depth camera, color/depth image processing, symmetry-based computer vision, and 3-D modeling and reconstruction.