Person Re-identification Based on DropEasy Method

HUIYANG WANG, TAO FANG, YINGLE FAN, AND WEI WU
Laboratory of Pattern Recognition and Image Processing Hangzhou Dianzi University, Hangzhou 310018, China
Corresponding author: Yingle Fan (fan@hdu.edu.cn)

ABSTRACT: Current major person re-identification impose network constraints by Dropout regularization and make features more independent by adopting the random zero features method. However, such random zero regularization methods are defective in promoting network performance because it does not take into account that different features contribute differently to network performance. To make indiscriminative features more valued in network training, a person re-identification method based on DropEasy method is put forward in this paper. Features are classified into discriminative ones and indiscriminative ones according to calculate the distance between feature vectors of positive or negative sample pairs, wherein the discriminative features are zeroed while the indiscriminative features are reserved, and the network only learns through indiscriminative features. Because a network is inclined to complete information by surrounding features in feature maps, where Dropout plays no part, DropEasy2d that can be effectively applied to convolutional layers is further presented in this paper, which can search for discriminative feature areas in the feature map by sliding windows and zero them to effectively constrain network learning. Extensive experiments show that DropEasy and DropEasy2d work better than Dropout in regularizing networks. In the Market-1501 dataset for instance, DropEasy can raise IDE's mAP accuracy and Rank-1 accuracy to 72.7% (+8.8%) and 90.5% (+6.8%) respectively, and DropEasy2d can raise IDE's mAP accuracy and Rank-1 accuracy to 68.5% (+4.6%) and 88.7% (+5.0%) respectively.

INDEX TERMS: Person Re-identification; Regularization; DropEasy; Dropout; Convolutional Neural Network (CNN)
SECTION I. 
Introduction

Person re-identification is a crucial branch of computer vision, which is typically used for the person matching task in a cross-camera scenario. Whereas person re-identification is conducive to public safety supervision and criminal investigation among others, it has received attention from scholars and research institutions day by day in recent years. However, how to effectively improve the accuracy of person re-identification remains a challenging work due to a difference in the camera angle, complicated scenarios, and ever-changing person poses among other influence factors.

Current major person re-identification studies are mainly concerned with methods based on feature learning. Local maximal occurrence (LOMO) [1], local binary patterns (LBP) [2], and invariant feature transform (SIFT) [3], to name a few, are typical methods for manually extracting features. In the wake of the development of deep learning, the accuracy of person re-identification has been improved substantially. Qian et al. [4] proposed a pose normalization method to get insensitive pose features; Zhong et al. [5] put forward a camera style switch method to get insensitive camera angle features; Qi et al. [6] used a mask-guided method to get features immune from background disturbance; Sun et al. [7] advised to divide features into sub-features and learn such sub-features one by one to get more discriminative local features; Zhao et al. [8] localized local body parts through pose estimation to learn features of the parts; Liu et al. [9] and Bai et al. [10] introduced LSTM [11] in person re-identification to learn horizontal local features; Li et al. [12] used a spatial transform network (STN) [13] and Zhang et al. [14] calculated the minimum distance of each local branch to align with body parts to get alignment features; Li et al. [15] promoted the network feature learning ability in an attention-based mechanism; Kalaveh et al. [16] granted a person’s every part with weight through a semantic analysis to better learn feature expression.

Apart from methods based on feature learning, methods based on metric learning are more valued than before. Cross-view quadratic discriminant analysis (XQDA) [1], for instance, is a typical traditional metric learning method. Later deep learning is blended into metric learning methods. Varior et al. [17] used contrastive loss [18] in Siamese network training. Triplet loss has been used for network training in some work [20][21][22] since it was put forward by Weinberger et al. [19]. Later the triplet loss was modified by Hermans et al. [23] by setting the margin between a maximum distance of positive sample pair and a minimum distance of negative sample pair, as a result of which, triplet loss has become one of the most frequently used loss functions in person re-identification in recent years. Chen et al. [24] raised quadruplet loss in view of the absolute distance between positive and negative samples, lowering intra-class variance while increasing the inter-class variance. Wen et al. [25] came up with an idea of center loss to address the problem of no constraint in the Softmax loss intra-class distribution. Fan et al. [26] took a different way—when using Softmax function, getting the inner product by multiplying the norm of vectors by cosine value so that a margin was added to the cosine value to effectively expand the inter-class distance. 

Both types of methods—ones based on feature learning and ones based on metric learning—use regularization to generalize networks but current regularization methods have their own evident defects. Dropout [27] for instance, zeros some feature vectors at random. But this random zero method does not take into account that different features contribute differently to network performance. Features extracted from a network actually fall into discriminative features and indiscriminative features, wherein the former can make it easy to differentiate persons of different identities or recognize persons of the same identity while the latter cannot. And it will be a better network if it has more discriminative features among total features extracted, hence how to turn indiscriminative features into discriminative features is a very meaningful challenge.

To address the preceding problem, a person re-identification based on DropEasy method is put forward in this paper. DropEasy resembles Dropout in the form of implementation. Their particular forms are shown in Figure 1. Through a distance contrast to a pair of feature vectors, DropEasy zeros discriminative features while maintaining indiscriminative features. More visually speaking, DropEasy collects some features to be noted in particular, which fail to recognize any particular person. Networks can merely rely on indiscriminative features to distinguish the similarity of persons in training. As a result, the networks need to extract more discriminative details.
Figure 1. $f^a = [x^a_1, x^a_2, x^a_3, x^a_4]$, $f^b = [x^b_1, x^b_2, x^b_3, x^b_4]$, wherein a deeper point color indicates a greater difference between corresponding features to $f^a$ and $f^b$, i.e., $|x^a_1 - x^b_1| < |x^a_2 - x^b_2| < |x^a_3 - x^b_3| < |x^a_4 - x^b_4|$. Dropout's working process is shown on the left, where 50% of features are zeroed at random. DropEasy's mechanisms of action on negative and positive samples are shown in the middle and on the right respectively. Where $f^a$ and $f^b$ fall into a negative sample pair (middle), $x^a_3, x^a_4$ and $x^b_3, x^b_4$ will be discriminative features to be zeroed; where $f^a$ and $f^b$ fall into a positive sample pair (right), $x^a_1, x^a_2$ and $x^b_1, x^b_2$ will be discriminative features to be zeroed.

From indiscriminative features to turn indiscriminative features into discriminative features step by step. The main contributions in this paper are summarized as follows:

1. The features of person re-identification network output are divided into two categories: discriminative features and indiscriminative features. This paper improves the regularization method of random zero features and proposes the DropEasy method, which is to zero the discriminative features while retaining the indiscriminative features, constrained networks only improve the distinguishing ability of features through indiscriminative feature learning.

2. Considering that if the regularization method of random zero is applied to the features on the feature map, the state of zero-discrete will be caused, and the features on the feature map have better correlation spatially. So the network is inclined to complete information by non-zero features, which leads to the weakening of the regularization. Therefore, this paper further proposes DropEasy2d according to the DropEasy method. By sliding the window to find indiscriminative features rectangle area on the feature map and zeroing it, it can effectively improve the network learning.

3. This method is applied to person re-identification networks such as IDE, SVDNet, PCB, etc. It can improve performance of these networks. At the same time, it is compared with the regularization method of random zero, which proves that this method can improve network to extract more discriminative features.

SECTION II.
Relevant Work
For too many network training parameters and lack of training data among other reasons, overfitting remains an inevitable problem during network training. Solutions to the preceding problem include data augmentation at network input and regularization of output in the intermediate layer of a network, wherein the latter is more prevalent. Dropout and DropConnet[28], for example, are widespread regularization methods, wherein the former zeros the output of each network node with a certain probability at random while the latter zeros each input weight connected to a node with a certain probability at random. However, features in the feature map output by CNN[29] are highly relevant to each other. As a result, if the zeroing features are too discrete, the network will tend to complete the information through non-zero features, where regularization plays no part. Therefore, such regularization methods as Dropout and DropEasy are used in full connected (FC) layers in most cases. To solve the existing problems in the preceding methods, Dropout, for example, DropPath [30] was put forward, which zeros the output of each sub-network to prevent the co-adaptation of other parallel sub-networks to make each sub-network more independent. However, by zeroing the output of a certain sub-network, this method puts an end to update any weight in the sub-network, indicating a lower network learning efficiency. And, this method can only be used in networks with a fractal structure [30]. Compared with DropPath, Shake-Shake[31] is more general. This method controls the output of multiple branches via random varying scale factors, which equal to synthesizes augmented features in every forward propagation and adjusts learning rate by random weight to do a reasonable disturbance to learning in back propagation. But this method is restricted to multi-residual branch networks like ResNetxt[32]. As an extension of Shake-Shake, ShakeDrop[33] applies to single-residual branch networks, which controls the output of residual branches through a combination of multiple random variables to zoom the branches but is restricted to networks with at least one residual branch. Targeting 2D feature maps, SpatialDropout[34] zeros a certain dimension at random. Such a structural zero method can address the problem of discretely zeroing features exist in Dropout and apply to Convolution networks. DropBlock[35] is an extension of SpatialDropout. Their only difference is as follows: SpatialDropout zeros strip areas whose length or width is 1 while DropBlock zeros rectangle areas. In view of the horizontal and vertical adaptation effects, DropBlock has better regularization effects than SpatialDropout.

It is found that the preceding regularization methods adopt a random method to equate all the output features but in actual training, a priority should be given to indiscriminate features in the network for its ability of feature recognition can be effectively enhanced by turning indiscriminate features into discriminative features through training. Based on the preceding idea, DropEasy proposed in this paper is an enhanced Dropout, which applies to and works out in both FC layers and convolutional layers. Used in a convolutional layer, DropEasy finds and zeros discriminative features in a rectangle area on the feature map. In a word, the method in this paper can generalize a person re-identification network mainly in line with the fundamental idea of dropping indiscriminate features while preserving indiscriminate features.

SECTION IV.

Method Proposed

DropEasy bears a similarity with Dropout, both of which zero some features. But when it comes to the specific form of implementation, they vary from each other to a great extent. Dropout zeros some features in the feature vector $f$ at random and reduces the dependence between features to generalize the network; DropEasy mines and zeros discriminative features between positive or negative sample pairs to get the indiscriminate feature vector $\bar{f}$ used to learn the difference between positive and negative sample pairs for network constraining, which can get more discriminative features. The pseudocodes for using DropEasy are set out in Algorithm 1. DropEasy has only one hyper-parameter $p$ intended to control the zero rate of feature vectors. To be specific, the higher $p$ is, the more features will be zeroed in $\bar{f}$ and the harder the network learning will be; on the contrary, the lower $p$ is, the fewer features will be zeroed in $\bar{f}$ and the easier the network learning will be.

For a differentiation from DropEasy that applies to FC layers, DropEasy that applies to convolutional layers is referred to DropEasy2d in this paper, which has a different form of implementation from that applies to FC layers. Whereas if features in the feature map are zeroed in line with rules in Algorithm 1, zeroing features will be discrete; features in the feature map relate to each other, hence an association

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/.
can be made in the network by non-zero features around the zeroing features, weakening effects of DropEasy2d. Inspired by DropBlock, DropEasy2d need to mine a discriminative rectangle area in the feature map. To be specific, differences of features in sliding windows are counted up in this paper to mine a discriminative feature area. The pseudocodes for its specific operation process are shown in Algorithm 2. DropEasy2d has two hyper-parameters ($M_h$ and $M_w$) separately used for control of a length and a width of a mask area. $M_h$ and $M_w$ are in direct proportion to the zero rate in the feature map and difficulty of network learning and vice versa. An example of DropEasy2d is given in Figure 2.

Algorithm 1

**Input:** $f^a$, the feature vector of sample $a$ of dim $N$ 
$p$, zero ratio 
$l$, the label of sample pair a,b 

If not training then 
$f_h^a = f^a$, $f_h^b = f^b$ 
end if 

$d = \text{abs}(f^a - f^b)$ 
$M = \text{int}(p \times N)$ 
$mask = \text{ones}(N)$ 

if $l == 1$ then 
$\text{index} = \text{argsort}(d, \text{from small to big})$ 
else if $l == 0$ then 
$\text{index} = \text{argsort}(d, \text{from big to small})$ 
end if 

$\text{mask}[\text{index}] = 0$ 
$f_h^a = f^a \times \text{mask}/(1 - p)$, $f_h^b = f^b \times \text{mask}/(1 - p)$ 

return $f_h^a \cdot f_h^b$

Algorithm 2

**Input:** $F^a$, feature maps of sample $a$ of size $C \times H \times W$ 
$f$, feature vector of sample $b$ of dim $N$ 
$l$, label of sample pair a,b 

$M_h$, the height of the mask 
$M_w$, the width of the mask 

If not training then 
$\overline{F^a} = \overline{F^a}$, $\overline{F^b} = \overline{F^b}$ 
end if 

$mask = \text{ones}(C, H, W)$ 
$D = |\text{sum}(F^a, \text{along channel}) - \text{sum}(F^b, \text{along channel})|$ 

if/ else if $l == 1$ then 
find $i_h, i_w$ make $\text{sum}(D[i_h: i_h + M_h, i_w: i_w + M_h]) \leq \text{sum}(D[k: k + M_h, j: j + M_h]) \ \forall k \in \{0: H - M_h\}, \forall j \in \{0: W - M_w\}$ 
else if $l == 0$ then 
find $i_h, i_w$ make $\text{sum}(D[i_h: i_h + M_h, i_w: i_w + M_h]) \geq \text{sum}(D[k: k + M_h, j: j + M_h]) \ \forall k \in \{0: H - M_h\}, \forall j \in \{0: W - M_w\}$ 
end if 

$\text{mask}[:, i_h: i_h + M_h, i_w: i_w + M_w] = 0$ 
$\overline{F^a} = f^a \times \text{mask}/(1 - M_h \times M_w/H \times W)$, $\overline{F^b} = f^b \times \text{mask}/(1 - M_h \times M_w/H \times W)$ 

return $\overline{F^a} \cdot \overline{F^b}$
**Figure 2. Process Diagram of DropEasy2d.** Deeper colors represent a greater difference between features of sample pairs in the feature map. Specifically speaking, the first line shows positive sample pairs, where the yellow area has the minimum sum of feature differences, hence the features are recognized as discriminative features to be zeroed; the second line shows negative sample pairs, where the yellow area has the maximum sum of feature differences, hence such features need to be zeroed.

**SECTION VI. Experimental**

**A. Experimental Data**

In this paper, an experimental test is made in 3 person re-identification datasets: Market-1501[36], DukeMTMC-reID[37][38] and CUHK03[39]. Data in Market-1501 and DukeMTMC-reID are accessed by 6 cameras and 8 cameras respectively; data in each ID of CUHK03 are accessed by 2 cameras. In addition, CUHK03 comprises two types of datasets: CUHK03-labeled for manually marking person boxes and CUHK03-detected that marks person boxes in the DPM algorithm. The original CUHK03 is divided into 20 training sets and testing sets at random for cross-validation, which applies to the method of manually extracting features in most cases. But in the CUHK03 training and testing in this paper, a new protocol[40] is used. Table 1 gives an overview of data in 3 datasets.

<table>
<thead>
<tr>
<th>Data</th>
<th>Market-1501</th>
<th>DukeMTMC-reID</th>
<th>CUHK03</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ID</td>
<td>Image</td>
<td>ID</td>
</tr>
<tr>
<td>Train</td>
<td>751</td>
<td>12936</td>
<td>702</td>
</tr>
<tr>
<td>Query</td>
<td>1433</td>
<td>24416</td>
<td>702</td>
</tr>
<tr>
<td>Gallery</td>
<td>751</td>
<td>15913</td>
<td>1110</td>
</tr>
</tbody>
</table>

**B. Experimental Settings**

The method proposed in this paper is verified in an IDE[41] network. IDE’s original structure is shown in Figure 3, where Resnet-50[42] used in ImageNet[43] for preliminary training acts as IDE’s backbone. Two measures are then added to IDE for optimization in this paper: (1) batch normalization of the feature vector \( f_1 \); (2) batch normalization of the embedded feature vector \( f_2 \). As a result, IDE’s network convergence is faster with a better performance than the original IDE. To be specific, IDE’s mAP accuracy and Rank-1 accuracy hit 63.9% and 83.7% in the Market-1501 dataset.

Training and testing are carried out in this paper by a 1080ti GPU. During the training, random horizontal flipping and normalized processing are done to input images whose size is adjusted to 256x128. In this paper, the network is trained in the momentum-based gradient descent method, wherein the momentum is fixed at 0.9. The initial learning rates of pretrained network layers and other network layers are 0.01 and 0.1 respectively, and each learning rate decreases to 1/10 of its initial value per 8k iterations. The whole training procedure has 12k iterations. In each iteration, 32 images with different IDs are selected at random to fall into Group A and another 32 images are selected at random to fall into Group B, wherein only half of the images in Group B bear an ID correspondence with ones in Group A, hence the training involves 64 batches.

In evaluation, the Euclidean distance between each query image and other camera gallery images is worked out and such distance values are sorted in a descending order to arrive at a cumulative matching characteristic (CMC) curve. In this paper, the evaluation is made on single-query settings only, where average mAP accuracy and Rank-1 accuracy are calculated as evaluation indicators on the premise of no reranking[40].
Figure 3. IDE’s original structure. First of all, a $2048 \times 8 \times 4$ feature map is output, where Resnet-50 acts as the backbone. Subsequently, $f_1$ and $f_2$ are obtained through the Global AvgPooling and Embedded FC layer respectively. Finally, $f_2$ is used to output prediction results in the Classifier FC layer. During the training, Softmax loss applies.

C. Experimental Evaluation

1) Evaluation on DropEasy and DropEasy2d Methods

Table 2 shows the evaluation on IDE+DropEasy in 3 datasets. During the training, DropEasy is added after $f_1$ and $p$ is set to 0.7,0.7 and 0.8 in Market-1501,DukeMTMC-reID and CUHK03, respectively. It is found that DropEasy drives an increase of IDE’s mAP accuracy in Market-1501, DukeMTMC-reID, CUHK03-labeled and CUHK03-detected from 63.9%, 50.1%, 31.9%, and 29.7% to 72.7% (+8.8%), 58.9% (+8.8%), 46.4% (+14.5), and 44.8% (+15.1%). The higher accuracy indicates a better network performance, which also means the network extracts more discriminative features, both of which are DropEasy’s greatest strengths. In addition, IDE has a higher mAP accuracy and Rank-1 accuracy at Market-1501 and DukeMTMC-reID where $f_1$ instead of $f_2$ acts as the output feature, indicating that DropEasy has a greater impact on $f_1$ that plays an immediate part; in CUHK03, however, $f_2$ makes a better performance than $f_1$. Maybe because $f_2$ combines $f_1$ with higher discriminating ability, applying to more challenging datasets as CUHK03.

Figure 4 shows person matching examples. It is found that in spite of a great difference between gallery images and each query image regarding the angle of view, which means existing many indiscriminative features, matching works out in the IDE where DropEasy is used, indicating that DropEasy can turn indiscriminative features into discriminative features indeed to improve the ability of feature recognition.

Table 2. $f_1$ vs. $f_2$ as output feature in IDE

<table>
<thead>
<tr>
<th>Method</th>
<th>Market-1501 mAP</th>
<th>Rank-1</th>
<th>DukeMTMC-reID mAP</th>
<th>Rank-1</th>
<th>CUHK03_labeled mAP</th>
<th>Rank-1</th>
<th>CUHK03_detected mAP</th>
<th>Rank-1</th>
</tr>
</thead>
<tbody>
<tr>
<td>IDE</td>
<td>63.9%</td>
<td>83.7%</td>
<td>50.1%</td>
<td>71.7%</td>
<td>31.9%</td>
<td>35.9%</td>
<td>29.7%</td>
<td>32.2%</td>
</tr>
<tr>
<td>IDE+DropEasy, $f_1$</td>
<td>72.7%</td>
<td>90.5%</td>
<td>58.9%</td>
<td>78.6%</td>
<td>45.8%</td>
<td>49.9%</td>
<td>44.0%</td>
<td>46.4%</td>
</tr>
<tr>
<td>IDE+DropEasy, $f_2$</td>
<td>72.1%</td>
<td>89.8%</td>
<td>58.0%</td>
<td>77.9%</td>
<td>46.4%</td>
<td>50.9%</td>
<td>44.8%</td>
<td>47.0%</td>
</tr>
</tbody>
</table>
Table 3. $f_1$ vs. $f_2$ as output feature in IDE

<table>
<thead>
<tr>
<th>Method</th>
<th>Market-1501 mAP Rank-1</th>
<th>DukeMTMC-reID mAP Rank-1</th>
<th>CUHK03_labeled mAP Rank-1</th>
<th>CUHK03_detected mAP Rank-1</th>
</tr>
</thead>
<tbody>
<tr>
<td>IDE</td>
<td>63.9% 83.7%</td>
<td>50.1% 71.7%</td>
<td>31.9% 35.9%</td>
<td>29.7% 32.2%</td>
</tr>
<tr>
<td>IDE+DropEasy2d,$f_1$</td>
<td>68.5% 88.7%</td>
<td>54.6% 75.9%</td>
<td>41.7% 45.5%</td>
<td>39.3% 43.1%</td>
</tr>
<tr>
<td>IDE+DropEasy2d,$f_2$</td>
<td>67.4% 87.2%</td>
<td>54.0% 74.2%</td>
<td>41.9% 46.1%</td>
<td>39.7% 43.5%</td>
</tr>
</tbody>
</table>

Figure 4. Here show some IDE data matching results in Market-1501 where DropEasy's $p$ value is 0.7, which are merely the top 5 sorting data. Besides, query images and gallery images are accessed by different cameras. Green boxes represent correct matching results while red boxes represent wrong matching results.

2) Hyper-parameter Design Experiment

Zero rate $p$ is a hyper-parameter that influences the performance of DropEasy. Figure 5 shows the performance of DropEasy in the wake of the variation of the zero rate $p$. Visually it is found that different datasets have different best $p$ values. In Market-1501, the best $p$ value is 0.7 and accordingly IDE's mAP accuracy and Rank-1 accuracy are 72.7% and 90.6% respectively; in CUHK03_labeled, the best $p$ value is 0.8 and accordingly IDE's mAP accuracy and Rank-1 accuracy are 46.4% and 51.6% respectively. And a higher zero rate $p$ means that the network is focused on more challenging indiscriminative features to better promote its ability of recognizing indiscriminative features provided no deviation in network learning, as a result of which, indiscriminative features will be turned into discriminative features step by step; however, if the zero rate $p$ is so high as to lose more key information and cause a deviation in network learning, it will lower the network performance and even eliminate convergence. Besides, Figure 6(a) shows that where DropEasy is not used in IDE, losses decrease faster with a small range of swing and converge after nearly 2k iterations. In contrast, where DropEasy is used in IDE, as the $p$ value rises, network loss decrease slows down with a greater range of swing and loss convergence is more challenging, as a result of which, learning time is prolonged. Figure 6(b) shows given the same $p$ value, compared with DropEasy, loss of network with Dropout decreases faster and range of swing is less, indicating that the method of mining and zeroing discriminative features has a greater disturbance to network learning than the method of random zero, which requires a better ability of feature extraction in the network through long-time learning. Therefore, selection of the best $p$ value is subject to particular data, which should facilitate quick convergence and enhance the network’s learning ability.

For DropEasy2d, selection of a proper $M_h$ and $M_w$ also matters very much. Figure 7 shows the impact of the variation of $M_h$ on DropEasy2d given a constant $M_w$ value of 7. The feature map of stage 3 output in IDE is in the size of 16x8. When $M_w$ is 7, the best value of $M_h$ will be 6 and the zero rate will approximate 1/3(6*7/16*8), and IDE's mAP accuracy and Rank-1 accuracy in CUHK03_labeled dataset will be 41.7% and 45.5% respectively. However, if $M_h$ is higher than 8, DropEasy2d will weaken IDE's original performance. Because lots of key features in the feature map are lost and the remaining features do not suffice for network learning and are seen as an excessive disturbance to network learning, leading to a learning deviation.
Figure 5. (a) and (b) show tests in Market-1501 and CUHK03_labeled respectively. The abscissa represents the zero rate $p$ while the ordinate represents the mAP accuracy and Rank-1 accuracy.

Figure 6. The abscissa shows the iterations done while the ordinate shows losses. (a) shows a comparison on the DropEasy loss decrease under different $p$ values; (b) shows a comparison on the Dropout and DropEasy loss decreases under the same $p$ value. Both (a) and (b) test on the Market-1501 dataset.

Figure 7. Variation of mAP accuracy and rank-1 accuracy in IDE where DropEasy2d applies in the wake of $M_h$ variation under a constant $M_w$ value of 7. this experiment is done on the CUHK03_labeled dataset.

3) Comparison with Other Regularization Methods

Dropout addresses the problem of features depending on each other by zeroing features at random, which can also be used on 2D feature maps; similar to Dropout, DropConnect differs from Dropout in zeroing input weights of features at random; DropBlock gives a consideration to the connection of 2D feature maps and zeros features in a random rectangle area; from the perspective of significance of features, DropEasy makes a movement to the zeroing rule (dropping discriminative features while reserving discriminative features) and hence works better than other regularization methods. Through analysis of the experiment in 4.3.2, it is found that DropEasy has a defect of slow convergence but arrives at the best zero rate to greatly promote the network performance. So does DropEasy2d.

DropEasy vs. Dropout. Given that the zero rate is 0.8 at FC layers, DropEasy's mAP accuracy and Rank-1 accuracy are 13.4% and 13.5% higher than those of Dropout. Such a great gap is mainly attributable to low performance of the IDE model where many indiscriminative features are involved in features extracted. Dropout can make features more...
independent but fails to effectively turn indiscriminative features into discriminative features. **DropEasy vs. DropConnect.** Given that the zero rate is 0.8 at FC layers, DropEasy's mAP accuracy and Rank-1 accuracy are 12.7% and 12.4% higher than those of DropConnect. DropConnect has the same defect as Dropout, hence logically DropEasy has a much better performance than DropConnect.

**DropEasy2d vs. Dropout2d.** At a convolutional layer, the zero rate is set as 0.3 for both the Dropout2d and DropEasy2d. It is found that Dropout2d can barely promote the IDE network. Therefore, given a low zero rate, Dropout2d has no regularization effects, for the network will complete the information of discrete zeroing features by non-zero features as referred to in Section II. DropEasy2d, however, does not have this defect through structural zeroing.

**DropEasy2d vs. DropBlock.** $M_w$ and $M_h$ are set as 6 and 7 for both methods at a convolutional layer. It is found that DropEasy2d's mAP accuracy and Rank-1 accuracy are 1.1% and 2.7% higher than those of DropBlock. Compared with DropBlock, DropEasy2d makes a better performance merely by an alteration of the zeroing rule, indicating that it will be more effective to use DropEasy2d than DropBlock in a 2D feature map.

Figure 8 gives a visual explanation to DropEasy and Dropout. It is found that after DropEasy is used, the network is inclined to focus on a smaller area with a small increase in the intra-class distance but a sharp increase in the inter-class distance. As a matter of fact, it is no exception, which is also found in other sample pairs. In this paper, both the small increase in the intra-class distance and the marked increase in the inter-class distance are considered to be on account of more focus of DropEasy on details, including significant ones learned from the only indiscriminative features during training.

4) Experiment of the Method in This Paper in Other Networks

Table 5 shows the comparison in all respects of network performance between before and after the method in this paper is introduced to the Market-1501 dataset. Where the method proposed in this paper is not used, IDE has a poor performance; after DropEasy2d is introduced, IDE's Rank-1 accuracy hits 88.7%, no lower than that of DPFL[45]. And after DropEasy is introduced, IDE's Rank-1 accuracy hits 90.5%, approximating that of HACNN[15]. After DropEasy is introduced to PCB[7] seen as one of the best networks so far, its mAP accuracy and Rank-1 accuracy will further rise to 78.3% (+0.9%) and 93.8% (+1.5%) respectively, indicating that the method in this paper can also promote a high-performance network. However, PCB learns 6 strip-typed local features and is restricted to learn in 6 local areas, hence it has less gain than IDE. Table 6 shows comparison results in the CUHK05_labeled dataset. It is found that in such classical person re-identification networks as SVDNet, DropEasy can raise the mAP accuracy and Rank-1 accuracy by 10.3% and 12.4% respectively, which seems more effective than Random Eraser. Random Eraser is recognized as one of the most powerful data augmentation methods but falls into a random zero method in nature merely for direct image input. IDE's performance can be further promoted by the DropEasy+DropEasy2d combination. What needs to be pointed out is that the DropEasy+DropEasy2d combination tends to eliminate convergence and even make a poorer performance than sold methods during training, hence hyper-parameter settings need to be noted in particular.

**Table 4.** The methods in This Paper vs. Other Regularization Methods in the IDE Network. Given a $p$ value of 0.8, $f_1$ applies to Dropout, DropEasy and DropConnect; the $p$ value is set as 0.3 for Dropout2d and $M_w$ and $M_h$ are set as 6 and 7 which apply to stage 3 for DropBlock and DropEasy2d. This experiment is done in the CUHK05_labeled dataset.

<table>
<thead>
<tr>
<th>Method</th>
<th>mAP</th>
<th>Rank-1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dropout</td>
<td>32.4%</td>
<td>36.4%</td>
</tr>
<tr>
<td>DropConnect</td>
<td>33.1%</td>
<td>37.5%</td>
</tr>
<tr>
<td>DropEasy</td>
<td><strong>45.8%</strong></td>
<td><strong>49.9%</strong></td>
</tr>
<tr>
<td>Dropout2d</td>
<td>32.3%</td>
<td>34.4%</td>
</tr>
<tr>
<td>DropBlock</td>
<td>40.6%</td>
<td>42.8%</td>
</tr>
<tr>
<td>DropEasy2d</td>
<td><strong>41.7%</strong></td>
<td><strong>45.5%</strong></td>
</tr>
</tbody>
</table>
SECTION V.

Conclusion

Regularization methods based on random zero cannot effectively promote network performance. To overcome this defect, a DropEasy zero method is put forward in this paper on the ground that features output from a person re-identification network are divided into discriminative features and indiscriminative features. Based on Euclidean distance comparison of feature vectors or feature maps of positive (negative) samples, farther (nearer) features are zeroed while farther (nearer) features are reserved and output. The person re-identification network keeps learning by the screened indiscriminative features to arrive at more detailed and discriminative features. To verify the method in this paper, an experiment is done in such person re-identification datasets as Market-1501, DukeMTMC-reID and CUHK03. In the first place, a network performance comparison is made between before and after the method applies to all networks. Results show that the
method can promote the performance of all networks. Besides, the method is compared with other regularization methods based on random zero to verify the method's superiority over other methods; last but not least, an experiment is designed where all kinds of hyper-parameters are involved and intended to reveal the impact of hyper-parameters on the method. Results show that optimal hyper-parameters need to be set subject to different networks and datasets. Through analysis on the preceding experimental results, the DropEasy regularization method is substantiated to be a new but effective method, offering a new idea for preventing network overfitting. DropEasy is expected to apply to other tasks instead of merely person re-identification through further optimization.

REFERENCES


This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/.